0230

Atty. Dkt. No. 043034/0158

*IN THE UNITED STATES PATENT AND TRADEMARK OFFICE*

#3
12-28-00

Applicant: Hidehito KUBO

Title: LOAD BALANCING METHOD AND SYSTEM BASED ON ESTIMATED ELONGATION RATES

Appl. No.: 09/680,517

Filing Date: 10/06/2000

Examiner: Unassigned

Art Unit: Unassigned



**CLAIM FOR CONVENTION PRIORITY**

Assistant Commissioner for Patents
Washington, D.C. 20231

Sir:

The benefit of the filing date of the following prior foreign application filed in the following foreign country is hereby requested, and the right of priority provided in 35 U.S.C. § 119 is hereby claimed.

In support of this claim, filed herewith is a certified copy of said original foreign application:

- Japanese Patent Application No. 11-285271 filed October 6, 1999.

Respectfully submitted,

Date November 28, 2000

By Phillip J. Articola
for / David A. Blumenthal
Attorney for Applicant
Registration No. 26,257

Reg. No. 38,819

FOLEY & LARDNER
Washington Harbour
3000 K Street, N.W., Suite 500
Washington, D.C. 20007-5109
Telephone: (202) 672-5407
Facsimile: (202) 672-5399

F05-489



# 日本国特許庁
## PATENT OFFICE
## JAPANESE GOVERNMENT

別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office.

出願年月日
Date of Application: １９９９年１０月　６日

出願番号
Application Number: 平成１１年特許願第２８５２７１号

出願人
Applicant (s): 日本電気株式会社

２０００年　７月１４日

特許庁長官
Commissioner,
Patent Office 及川耕造

出証番号　出証特２０００－３０５５２１４

【書類名】　特許願

【整理番号】　33509609

【提出日】　平成11年10月 6日

【あて先】　特許庁長官殿

【国際特許分類】　G06F 15/16

【発明者】

【住所又は居所】　東京都港区芝五丁目７番１号　日本電気株式会社内

【氏名】　久保　秀士

【特許出願人】

【識別番号】　000004237

【氏名又は名称】　日本電気株式会社

【代理人】

【識別番号】　100088959

【弁理士】

【氏名又は名称】　境　廣巳

【手数料の表示】

【予納台帳番号】　009715

【納付金額】　21,000円

【提出物件の目録】

【物件名】　明細書　1

【物件名】　図面　1

【物件名】　要約書　1

【包括委任状番号】　9002136

【プルーフの要否】　要

【書類名】　明細書

【発明の名称】　推定伸長率に基づくトランザクション負荷分散方法及び方式並びにコンピュータ可読記録媒体

【特許請求の範囲】

【請求項１】　トランザクション処理要求を発生する端末装置群と、該要求の処理を負荷分担して実行する複数の計算機からなるシステムにおいて、

各計算機の負荷状況を推定し、該推定負荷状況に基づいてすべての計算機について処理時間の推定伸長率を求め、この推定伸長率をベースとして各計算機の負荷指標の値を計算し、該負荷指標の値に基づいてトランザクション実行の各計算機への配分を決定することを特徴とする推定伸長率に基づくトランザクション負荷分散方法。

【請求項２】　前記各計算機の負荷状況の推定に当たっては、一定時間ごとに各計算機の負荷データを測定してこれを元に各計算機の負荷状況を推定することを特徴とする請求項１に記載の推定伸長率に基づくトランザクション負荷分散方法。

【請求項３】　前記各計算機の負荷状況の推定に当たっては、各計算機におけるトランザクション処理の開始・終了に応じて各計算機の処理中トランザクション現在数を常に把握し、前記推定伸長率を求めるに当たっては、前記測定に基づいた推定負荷状況と前記処理中トランザクション現在数とに基づくことを特徴とする請求項２に記載の推定伸長率に基づくトランザクション負荷分散方法。

【請求項４】　端末からトランザクション処理要求が到着したときに、前記推定負荷状況に基づいてすべての計算機について前記処理時間の推定伸長率を求め、この推定伸長率をベースに各計算機の前記負荷指標の値を計算し、該到着した処理要求の実行に最適な計算機を該負荷指標の値から決定して処理させるようにすることを特徴とする請求項１、２または３に記載の推定伸長率に基づくトランザクション負荷分散方法。

【請求項５】　前記一定時間ごとに測定する各計算機の負荷データとして、処理中トランザクション数およびＣＰＵ系に滞在する業務処理プロセス数を測定して使用することを特徴とする請求項２、３または４に記載の推定伸長率に基づ

くトランザクション負荷分散方法。

【請求項６】　前記一定時間ごとに測定する各計算機の負荷データとして、処理中トランザクション数およびＣＰＵ使用率を測定して使用することを特徴とする請求項２、３または４に記載の推定伸長率に基づくトランザクション負荷分散方法。

【請求項７】　前記各計算機の負荷状況の推定に当たって、一定時間ごとに測定した前記各計算機の負荷データの系列を総合的に用いて推定することを特徴とする請求項２、３、４、５または６に記載の推定伸長率に基づくトランザクション負荷分散方法。

【請求項８】　前記処理時間の推定伸長率を求めるにあたり、既に得られている前記推定負荷状況のデータを、前記処理中トランザクション現在数を用いて補正して使用することを特徴とする請求項３、４、５、６または７に記載の推定伸長率に基づくトランザクション負荷分散方法。

【請求項９】　前記計算機の負荷指標として、該計算機へ新規にトランザクションを割当てる前あるいは割当て後における、総推定伸長率、すなわち、該計算機における前記処理時間の推定伸長率に該計算機の処理中トランザクション数を乗じた値を用いることを特徴とする請求項１、２、３、４、５、６、７または８に記載の推定伸長率に基づくトランザクション負荷分散方法。

【請求項１０】　前記計算機の負荷指標として、該計算機へ新規にトランザクションを割り当てた後における前記総推定伸長率と、割当て前における前記総推定伸長率との差を用いることを特徴とする請求項１、２、３、４、５、６、７または８に記載の推定伸長率に基づくトランザクション負荷分散方法。

【請求項１１】　トランザクション処理要求を発生する端末装置群と該要求の処理を負荷分担して実行する複数の計算機からなるシステムにおいて、

各計算機の負荷状況を推定する負荷データ測定手段と、

推定した該負荷状況を記憶する負荷データ記憶手段と、

該推定負荷状況に基づいてすべての計算機について処理時間の推定伸長率を求め、この推定伸長率をベースとして各計算機の負荷指標の値を計算し、該負荷指標の値に基づいてトランザクション実行の各計算機への配分を決定する実行計算

機選択手段と、

前記各計算機ごとにその上に存在し、複数のトランザクション実行を並列に行い、前記実行計算機選択手段に指令されたトランザクションの実行を管理するトランザクション処理手段とを備えたことを特徴とする推定伸長率に基づくトランザクション負荷分散方式。

【請求項１２】　前記負荷データ測定手段は、一定時間ごとに各計算機の負荷データを測定してこれを元に各計算機の負荷状況を推定することを特徴とする請求項１１に記載の推定伸長率に基づくトランザクション負荷分散方式。

【請求項１３】　前記負荷データ測定手段は、各計算機におけるトランザクション処理の開始・終了に応じて各計算機の処理中トランザクション現在数を常に把握し、

前記実行計算機選択手段は、すべての計算機について前記処理時間の推定伸長率を求めるに際し、前記測定に基づいた推定負荷状況と前記処理中トランザクション現在数とに基づくことを特徴とする請求項１２に記載の推定伸長率に基づくトランザクション負荷分散方式。

【請求項１４】　前記実行計算機選択手段は、端末から前記トランザクション処理要求が到着したときに起動され、該到着した処理要求の実行に最適な計算機を決定して処理させるようにすることを特徴とする請求項１１、１２または１３に記載の推定伸長率に基づくトランザクション負荷分散方式。

【請求項１５】　前記負荷データ測定手段は、前記一定時間ごとに測定する各計算機の負荷データとして、処理中トランザクション数およびＣＰＵ系に滞在する業務処理プロセス数を測定し、負荷状況の推定に使用することを特徴とする請求項１２、１３または１４に記載の推定伸長率に基づくトランザクション負荷分散方式。

【請求項１６】　前記負荷データ測定手段は、前記一定時間ごとに測定する各計算機の負荷データとして、処理中トランザクション数およびＣＰＵ使用率を測定し、負荷状況の推定に使用することを特徴とする請求項１２、１３または１４に記載の推定伸長率に基づくトランザクション負荷分散方式。

【請求項１７】　前記負荷データ測定手段は、前記各計算機の負荷状況の推

定に当たって、一定時間ごとに測定した前記各計算機の負荷データの系列を総合的に用いて推定することを特徴とする請求項１２、１３、１４、１５または１６に記載の推定伸長率に基づくトランザクション負荷分散方式。

【請求項１８】　前記実行計算機選択手段は、各計算機の前記処理時間の推定伸長率を求めるにあたり、既に得られている前記推定負荷状況のデータを前記処理中トランザクション現在数を用いて補正して使用することを特徴とする請求項１３、１４、１５、１６または１７に記載の推定伸長率に基づくトランザクション負荷分散方式。

【請求項１９】　前記実行計算機選択手段は、各計算機の前記負荷指標として、該計算機へ新規にトランザクションを割当てる前あるいは割当て後における、総推定伸長率、すなわち、該計算機における処理時間の推定伸長率に該計算機の処理中トランザクション数を乗じた値、を用いることを特徴とする請求項１１、１２、１３、１４、１５、１６、１７または１８に記載の推定伸長率に基づくトランザクション負荷分散方式。

【請求項２０】　前記実行計算機選択手段は、各計算機の前記負荷指標として、該計算機へ新規にトランザクションを割り当てた後における前記総推定伸長率と、割当て前における前記総推定伸長率との差を用いることを特徴とする請求項１１、１２、１３、１４、１５、１６、１７または１８に記載の推定伸長率に基づくトランザクション負荷分散方式。

【請求項２１】　前記実行計算機選択手段と前記負荷データ記憶手段とがそれぞれシステムに一つだけ存在して集中的にその機能を実行し、前記実行計算機選択手段は各計算機の前記負荷指標の値を直接的に反映してトランザクションの配分を行うことを特徴とする請求項１１、１２、１３、１４、１５、１６、１７、１８、１９または２０に記載の推定伸長率に基づくトランザクション負荷分散方式。

【請求項２２】　前記実行計算機選択手段が各計算機ごとに一つずつ分散して存在し、固定的、静的または準静的な分配方式によって計算機に配分されてきたトランザクションについて、該計算機上の実行計算機選択手段が、すべての計算機の前記負荷指標の値に基づき、次の２つの決定すなわち、該計算機でそのま

ま処理するか他に回すかを閾値判断で決定、および他に回す場合はその送付先を決定、を行うことを特徴とする請求項１１、１２、１３、１４、１５、１６、１７、１８、１９または２０に記載の推定伸長率に基づくトランザクション負荷分散方式。

【請求項２３】　トランザクション処理要求を発生する端末装置群と、該要求の処理を負荷分担して実行する複数の計算機からなるシステムにおいて、各計算機の処理時間の伸長率を推定し、この推定伸長率をベースとした各計算機の負荷指標に基づいてトランザクション処理要求を各計算機へ配分することを特徴とする推定伸長率に基づくトランザクション負荷分散方法。

【請求項２４】　トランザクション処理要求を発生する端末装置群および該要求の処理を負荷分担して実行する複数の計算機に接続され、前記端末装置群からのすべての処理要求を集中的に受け取って前記計算機に配分する中継配分装置を構成するコンピュータに、

前記各計算機から一定時間ごとに通知される当該計算機のＣＰＵ系に滞在する業務処理プロセス数あるいはＣＰＵ使用率、および処理中トランザクション数を含む負荷データに基づき、各計算機の負荷状況を推定するステップ、

推定された負荷状況に基づいてすべての計算機について処理時間の推定伸長率を求めるステップ、

該推定伸長率をベースとして各計算機の負荷指標の値を計算するステップ、

該負荷指標の値に基づいてトランザクション実行の各計算機への配分を決定するステップ、

を実行させるプログラムを記録したコンピュータ可読記録媒体。

【請求項２５】　トランザクション処理要求を発生する端末装置群からの要求の処理を負荷分担して実行する複数の計算機のそれぞれに、

自計算機のＣＰＵ系に滞在する業務処理プロセス数あるいはＣＰＵ使用率、および処理中トランザクション数を含む負荷データを一定時間ごとに測定し、これを元に自計算機の負荷状況を推定して記憶すると共に他のすべての計算機に通知するステップ、

処理要求の到着を契機に、すべての計算機についての前記推定負荷状況から各

計算機の処理時間の推定伸長率、該推定伸長率をベースにした各計算機の負荷指標を求めるステップ、

該求めた各計算機の負荷指標の値に基づいて前記到着した処理要求を自計算機で処理すべきか他計算機に依頼すべきかを判断し、他に送付する場合はその送り先を決定し、選択した実行先に送付依頼するステップ、

を実行させるプログラムを記録したコンピュータ可読記録媒体。

【請求項２６】　トランザクション処理要求を発生する端末装置群からの処理要求を一括して受け取って静的／準静的な方式により当該処理要求を配分する計算機を決定する中継仮配分装置に接続された前記それぞれの計算機に、

自計算機のＣＰＵ系に滞在する業務処理プロセス数あるいはＣＰＵ使用率、および処理中トランザクション数を含む負荷データを一定時間ごとに測定し、これを元に自計算機の負荷状況を推定して記憶すると共に他のすべての計算機に通知するステップ、

処理要求の到着を契機に、すべての計算機についての前記推定負荷状況から各計算機の処理時間の推定伸長率、該推定伸長率をベースにした各計算機の負荷指標を求めるステップ、

該求めた各計算機の負荷指標の値に基づいて前記到着した処理要求を自計算機で処理すべきか他計算機に依頼すべきかを判断し、他に送付する場合はその送り先を決定し、選択した実行先に送付依頼するステップ、

を実行させるプログラムを記録したコンピュータ可読記録媒体。

【発明の詳細な説明】

【０００１】

【発明の属する技術分野】

本発明は、クラスタ構成などと呼ばれる比較的緊密に結合された複数の計算機が負荷分担してトランザクション処理を実行するシステムにおいて、トランザクション処理の負荷を各計算機に分散させる方式に関し、特に各計算機における負荷状況を示す指標である「処理時間の推定伸長率」に基づいて処理要求を動的に配分することにより計算機間の負荷をバランスさせ、全体として応答時間の平均及びばらつきを小さく保つ方式に関する。

【０００２】

【従来の技術】

この種の負荷分散方式は、複数の処理装置（計算機）をもつシステムにおいて、規模の小さい処理を要求するメッセージが次々に大量に到着するのを、これら複数の計算機に適切に分配することによって計算機間で負荷を分散させ、システムから最大の性能を引き出そうとする。個々の処理が小規模なので、一般に、処理の途中で移動させることは考えず、到着時に処理を実行すべき計算機を決定してそこへ送付し、終了までそこで処理させる。また、対話型処理なので、負荷分散の最終的な目標は応答時間の平均（及びばらつき）の最小化である。処理要求の到着時に実行させる計算機を決定するが、この実行計算機の選択に当たっては、基本的に、負荷が最も低い計算機を選ぶことになる。ここで、何を「負荷の指標」とするかという問題が生ずる。従来、負荷の指標としては、計算機のＣＰＵ使用率、実行中の処理数、近い過去の応答時間の実績などが、個々に単独で、あるいは組み合わせて用いられていた。

【０００３】

従来のシステムの一例が、特開平１０－３１２３６５号公報に記載されている（従来技術１と呼ぶ）。ここでは、一定時間ごとにサーバー（計算機）の負荷状態（実施例によればＣＰＵ使用率）を計測して格納しておき、処理要求到着時には格納してある負荷状態に基づき最も負荷の低いサーバーを実行計算機として決定する。また、端末側で応答時間を監視していて、これが所定値を越えているなら、経路変換（実行計算機の変更）要求を出す。ここでは、第一の負荷指標としてＣＰＵ使用率が用いられている。ＣＰＵ使用率はよい指標ではあるが、測定値は過去の一定時間間隔における平均値であり、その後にあった処理の開始終了などの影響を含まないこともあり、「現在の負荷」を表しているという意味での信頼性はあまり高くない。特に、動的な制御の下では、次の測定時までは同じデータが用いられるので、この間に到着した処理要求は負荷が最小であると判断された特定の一つのサーバーに集中的に送られることになり、負荷のシーソー現象を起こす可能性がある。第二の負荷指標として、実行途中の処理の応答時間の実績が、その処理自身の実行先切り替えの判断に用いられている。ここにおけるよう

に実行途中における実行計算機の切り替えがオーバヘッド小さく可能な場合があるとすれば、有効な指標であろう。しかし、到着時の配分においては、その処理自身の実績がないのでこの指標は使用不可能である。

【０００４】

従来のシステムの別の一例が、特開平１０－２７１６８号公報に記載されている（従来技術２と呼ぶ）。ここでは、各計算機で最後に実行を終了したメッセージについてその処理時間を記憶しておき、この時間にその計算機上で処理中のメッセージの数を乗じたものを負荷の指標として用いている。メッセージの到着時には、すべての計算機についてこの負荷指標の値を計算し、この値の最も小さい計算機にメッセージを送り処理を依頼する。この場合、最後に終了したメッセージの処理時間がその計算機上の処理時間を代表するか、という問題がある。この処理時間は、その計算機の混み具合とその最終終了メッセージ処理のジョブ特性（純処理時間、ＣＰＵ／入出力の比率）とを反映しているはずである。すべてのメッセージについて後者のジョブ特性が同一であるならば、負荷指標として目安を与えると考えられる。しかし、様々なジョブ特性のものが混在している一般の状況では、個々の処理時間実績をそのまま負荷状態を反映するものと考えると判断を誤る可能性が大きい。

【０００５】

他の従来のシステムの例が、特開平７－３０２２４２号公報に記載されている（従来技術３と呼ぶ）。ここには多くの請求項があるが、本発明に近いものは請求項９、段落番号「１５２」～「１６１」に記載されたものである。ここでは、トランザクション処理部の負荷を定期的に検出して時刻と共に負荷の履歴を記憶しておく。そして、負荷傾向Ｔｒを次の式で計算する。

Ｔｒ＝（Ｗ２－Ｗ１）／（Ｔ２－Ｔ１）

トランザクション処理要求を受けると、一定時間Ｔｉ後に処理負荷予測値が閾値Ｗｔを越えないと判断した場合（Ｔｒ・Ｔｉ≦Ｗｔ）、自分で受け付け、そうでなければ拒否する。あるいは、より負荷の低い他のサーバに処理を依頼する。この例では、負荷を定期的に検出してこれをベースとして判断するが、具体的に何をもって「負荷」とするかについては公報全体を通じて明確に規定されていない

。負荷の指標を規定することは負荷配分にとって重要な第一歩であるが、それがなされていない。また、Ｔｉ後の負荷を過去の線形外挿により予測しようとしているが（Ｔｒ・Ｔｉでは不足と思われるが、それは別として）、これは良い予測とは思えない。システム全体の負荷についてはこの種のマクロな予測も有効かも知れないが、自サーバについてはその先の負荷状況は、現在の状態と処理中トランザクションの終了タイミング、自分が新たに処理を受け入れるか否かというミクロな動きで決まるものであり、過去の傾向を延長してそのまま信じてしまうのは危険である。

【０００６】

また、上述の従来技術１，２，３はいずれも、到着したメッセージ自体の処理時間を最短にすることを狙って実行先を決定している。しかし、このような個別最適化がシステム全体としての最適化に直結するという保証は必ずしもあるわけではない。

【０００７】

他の従来のシステムの例が、1997年にSpringer社から発行されたOptimal Load Balancing in Distributed Computer Systems（H.Kameda他著）の第２２５頁～第２３２頁に記載されている（従来技術4と呼ぶ）。前提としているモデルを１台のＣＰＵに注目して示すと図２のようなものである。ジョブは到着すると、ＣＰＵ（図では計算機ｉ）とディスクの使用を繰り返し、処理を終了すると立ち去る。この間の時間が応答時間である。複数のジョブが並行処理されるので、ＣＰＵの前には待ち行列が生ずる。他のＣＰＵも、図示した計算機ｉと同じ位置づけになり、ディスクに対するアクセス時間はすべてのＣＰＵから同等である。ここでは、負荷指標として次の２つの式が示されている。

$$f_i = s_i (n_i + 1)^2 \qquad (式1)$$

$$F_i = s_i (n_i + 1) \qquad (式2)$$

ここで、ｆおよびＦは負荷指標、ｉは計算機番号、ｓはジョブのＣＰＵにおける純サービス時間の平均、ｎはＣＰＵ系に存在するジョブ数である。これらの式は、待ち行列理論で言う開放型待ち行列網モデルにおいて、ＣＰＵ系について平衡状態における平均値に関して成立する関係をもとに、平均応答時間をある意味で

最小化するという目的のために、小さいほどよい値として導き出されたものである。実際、（式２）はＣＰＵ系における平均滞在時間を表し、これに入出力系における平均滞在時間を加えると平均応答時間となるものである。これらの負荷指標は、静的負荷配分のための指標としてはある意味の最適性が証明されている。しかし、動的制御はその時々の状況に応じた制御を行いうるところにメリットがあり、ｓｉ，ｎｉについては、平衡状態における平均値でなく現在値を用いないと意味がない。現在値に関しｎｉは測定可能であるが、計算機ｉ上で実行中のジョブミックスの特性を反映するｓｉは直接には測定不可能である。当文献上における評価では、ｓｉとして全体の平均値を用いている。実行する処理が、ジョブ特性の観点から一種類でしかも特性のばらつきが小さいなら全体の平均値を用いてもよいであろうが、一種類でもばらつきが大きい場合や、現実には一般的と考えられる特性の異なる複数種類の処理が混在する場合には、全体の平均値を用いてしまうと動的制御のメリットが大きく失われることになる。

【０００８】

【発明が解決しようとする課題】

第１の問題点は、負荷分散のベースとなる各計算機の負荷状況の把握が不十分であるということである。従来は、処理中トランザクション数、ＣＰＵ使用率などがそれぞれ単独であるいは組み合わせて負荷指標として使われていたが、これらは、その時点で処理中のトランザクション群のＣＰＵ／入出力使用比率を含むジョブ特性まで含めた、システムの混み具合を充分に反映するものとは言えない。また、トランザクション処理のように小規模の処理要求が大量に到着するシステムでは、短い時間間隔で正確に負荷状況を把握する必要があるが、分散システムにおけるデータ収集のオーバヘッドを恐れ、収集頻度を少なくするような傾向があった。低オーバヘッドな良質のデータを用いる工夫と共に、クラスタ型などの環境ではデータ収集は高速・低オーバヘッドなので、これを生かして良い負荷分散を実現するような方式が求められる。

【０００９】

第２の問題点は、必ずしも、システム全体としての最適化（応答時間の平均、分散の最小化）を図るものではなかったということである。到着したトランザク

ションの配分先を決定するに際し、当該トランザクションにとってその時点で最適な（最短時間で処理できると予想される）計算機を選択していたが、このような個別最適化は、システム全体としての最適化につながることを、必ずしも保証するものではない。

【００１０】

【発明の目的】

本発明の目的は、クラスタ構成などと呼ばれる比較的緊密に結合された複数の計算機が負荷分担してトランザクション処理を実行するシステムにおいて、到着した処理要求に対し適切な実行先計算機を選択するための基準となる有効な負荷指標を提供し、これに基づく選択を小さいオーバヘッドで実行可能にすることにより、トランザクション処理の負荷を短期レンジでも計算機間でバランスさせ、もって、全体として応答時間の平均とばらつきを小さく保つことを可能にする動的な負荷分散方式を提供することにある。

【００１１】

【課題を解決するための手段】

本発明は、トランザクション処理要求を発生する端末装置群と該要求の処理を負荷分担して実行する複数の計算機からなるシステムにおいて、各計算機の処理時間の伸長率を推定し、この推定伸長率をベースとした各計算機の負荷指標に基づいてトランザクション処理要求を各計算機へ配分する。具体的には、本発明にかかる負荷分散方法にあっては、各計算機の負荷状況を推定し、該推定負荷状況に基づいてすべての計算機について処理時間の推定伸長率を求め、この推定伸長率をベースとして各計算機の負荷指標の値を計算し、該負荷指標の値に基づいてトランザクション実行の各計算機への配分を決定する。また、本発明にかかる負荷分散方式にあっては、各計算機の負荷状況を推定する負荷データ測定手段と、推定した該負荷状況を記憶する負荷データ記憶手段と、該推定負荷状況に基づいてすべての計算機について処理時間の推定伸長率を求め、この推定伸長率をベースとして各計算機の負荷指標の値を計算し、該負荷指標の値に基づいてトランザクション実行の各計算機への配分を決定する実行計算機選択手段と、前記各計算機ごとにその上に存在し、複数のトランザクション実行を並列に行い、前記実行計

算機選択手段に指令されたトランザクションの実行を管理するトランザクション処理手段とを有する。

【００１２】

各計算機の処理時間の伸長率とは、業務処理プロセスの応答時間、すなわち待ち時間も含む処理時間の、純処理時間（ＣＰＵ、ファイル装置という資源を実際に使用する時間の合計）に対する倍率を意味する。この伸長率は、当該計算機上で実行中のプロセスの集まり（ジョブミックス）の、動作中の群としてのプログラム特性（ＣＰＵ使用特性だけでなく、ＣＰＵ－Ｉ／Ｏ使用特性を含む）を反映している。従って、処理速度が同じ計算機ならば、同一の処理は伸長率の小さい計算機で実行した方が処理時間は短くなり、応答時間を短くできる。

【００１３】

各計算機における処理時間の伸長率は、一定時間ごとに各計算機の負荷データとして例えば処理中トランザクション数とＣＰＵ系に滞在する業務処理プロセス数、または、処理中トランザクション数とＣＰＵ使用率を測定し、これらに基づいて推定する。一定時間ごとに測定した負荷データの系列を総合的に用いて各計算機の負荷状況を推定したり、各計算機におけるトランザクション処理の開始・終了に応じて各計算機の処理中トランザクション現在数を常に把握しておき、この処理中トランザクション現在数を用いて推定負荷状況データを補正したりすれば、推定負荷状況の推定精度が高まり、ひいては伸長率の推定精度も高まる。

【００１４】

推定伸長率をベースとして各計算機の負荷指標の値を求め、これに応じて、到着する処理要求を計算機へスケジュールする。負荷指標としては、推定伸長率そのものを負荷指標とすることができる他、当該計算機へ新規にトランザクションを割当てる前あるいは割当て後における、総推定伸長率、すなわち、当該計算機における前記処理時間の推定伸長率に当該計算機の処理中トランザクション数を乗じた値を用いることができ、また、当該計算機へ新規にトランザクションを割り当てた後における前記総推定伸長率と、割当て前における前記総推定伸長率との差を用いることもできる。後者では、伸長率の増分最小の計算機が選択されるため、システム全体にとって当スケジュールによる応答時間総和の増加を最小に

する選択になる。

【００１５】

本発明の推定伸長率に基づくトランザクション負荷分散方式では、実行計算機選択手段と負荷データ記憶手段とがそれぞれシステムに一つだけ存在して集中的にその機能を実行し、前記実行計算機選択手段は各計算機の前記負荷指標の値を直接的に反映してトランザクションの配分を行うよう構成して良い。具体的には、すべての処理要求を集中的に受け取って計算機に配分する中継配分装置（図１の２）を備え、各計算機（図１の１ｘ）上に存在して一定時間ごとに負荷データを測定し中継配分装置に通知する手段（図１の１ｘ１と１ｘ３）と、これを受けて中継配分装置上で各計算機の負荷状況を推定して記憶すると共に各計算機の処理中トランザクション現在数を常に把握する手段（図１の８と６）と、中継配分装置上に存在し端末から処理要求が到着すると起動され、該到着処理要求を処理する計算機を決定して送付する実行計算機選択手段（図１の７）とを備え、実測に基づいたその時点の推定負荷状況と処理中トランザクション現在数とに基づいて各計算機の処理時間の推定伸長率を求め、この推定伸長率をベースに各計算機の負荷指標の値を計算し、該負荷指標の値から、到着した処理要求に関して動的負荷配分の観点から最適な計算機を決定して処理させるように動作する（この方式を第１の方式と呼ぶ）。

【００１６】

本発明の推定伸長率に基づくトランザクション負荷分散方式では、また、実行計算機選択手段が各計算機ごとに一つずつ分散して存在し、固定的、静的または準静的な分配方式によって計算機に配分されてきたトランザクションについて、該計算機上の実行計算機選択手段が、すべての計算機の前記負荷指標の値に基づき、次の２つの決定すなわち、該計算機でそのまま処理するか他に回すかを閾値判断で決定、および他に回す場合はその送付先を決定、を行うよう構成して良い。具体的な構成例としては、次の２つの方式が考えられる（それぞれ第２の方式、第３の方式と呼ぶ）。

【００１７】

第２の方式は、中継配分装置を備えず、したがって処理要求は静的な方式で各

計算機へ配分されるが、各計算機上に存在して一定時間ごとに負荷データを測定しこれを元に負荷状況を推定して記憶すると共に他のすべての計算機に通知する手段（図６の１ｘ１と８ｘ）と、処理要求の到着と共に到着計算機上で起動されてすべての計算機についての前記推定負荷状況から各計算機の処理時間の推定伸長率を求めこれを元に負荷指標の値を計算し、各計算機の該負荷指標の値に基づいて自計算機で処理すべきか他計算機に依頼すべきかを閾値を用いて判断し、他に送付する場合はその送り先を決定し、選択した実行先に送付依頼するよう動作する実行計算機選択手段（図６の７ｘ）とを備える構成である。

【００１８】

第３の方式は、処理要求を一括して受け取って静的／準静的方式により計算機に配分する中継仮配分装置（図７の２５）を備え、動的に最適とは言えない仮配分がなされるが、各計算機上に第２の方式と同一の、一定時間ごとに働く測定手段と、負荷状況推定手段、及び仮配分された処理要求の到着時に起動され各計算機の推定伸長率を計算しこれに基づいて負荷指標の値を求め、自計算機で処理すべきか他に依頼するとしたらどの計算機かを決定し処理を依頼するように動作する実行計算機選択手段（図７の７ｘ）を備える構成である。

【００１９】

【発明の実施の形態】

次に、本発明の実施の形態について図面を参照して詳細に説明する。

【００２０】

図１を参照すると、本発明の第１の実施の形態は、プログラム制御により動作する計算機群１と、中継配分装置２と、高速チャネル３と、ファイル装置群４と、端末装置群５と、通信網５１とから構成されている。

【００２１】

計算機群１は計算機１１～１ｎを含み、計算機１１～１ｎは、それぞれ、負荷データ測定手段Ａ１１１～１ｎ１と、トランザクション処理手段１１２～１ｎ２と、通信手段１１３～１ｎ３とを含み、トランザクション処理手段１１２～１ｎ２はそれぞれ複数の業務処理プロセスを含み（図示せず）、中継配分装置２は通信手段２１と、負荷データ記憶手段６と、実行計算機選択手段７と、負荷データ

測定手段Ｂ８とを含む。計算機１１～１ｎは主記憶を共有しないが、ファイル装置群４には高速チャネル３を介して性能的に同等の条件で接続されており、ファイル装置を共有している。中継配分装置２も高速チャネル３を介して計算機群１に接続され、端末群５は通信網５１を介して中継配分装置２に接続されている。

【００２２】

１つのトランザクションの処理の概略は次のようになる。トランザクション処理要求であるメッセージは端末群５に属する端末装置から送り出され、中継配分装置２に伝えられる。中継配分装置２は、受け取ったメッセージを処理する計算機１ｉを決定し、該計算機に高速チャネル３を介して該メッセージを送る。受け取った計算機１ｉではトランザクション処理を実行し、応答メッセージを作成して逆の経路を通して要求元端末に返す。

【００２３】

ここで、各計算機１１～１ｎに備わる負荷データ測定手段Ａ１１１～１ｎ１、トランザクション処理手段１１２～１ｎ２、通信手段１１３～１ｎ３をソフトウェア的に実現する場合、これらの各手段を実現するプログラムは図示しないＣＤ－ＲＯＭ、磁気ディスク、半導体メモリ等の機械読み取り可能な記録媒体に保存されており、計算機群１の立ち上げ時などに記録媒体に記録されたプログラムが各計算機に読み込まれ、各計算機の動作を制御することにより、各計算機上にこれら各手段を実現する。また、中継配分装置２に備わる通信手段２１、負荷データ記憶手段６、実行計算機選択手段７、負荷データ測定手段Ｂ８をソフトウェア的に実現する場合、これらの各手段を実現するプログラムは図示しないＣＤ－ＲＯＭ、磁気ディスク、半導体メモリ等の機械読み取り可能な記録媒体に保存されており、中継配分装置２を構成する計算機の立ち上げ時などに記録媒体に記録されたプログラムがその計算機に読み込まれ、その計算機の動作を制御することにより、その計算機上にこれら各手段を実現する。

【００２４】

上記の各手段はそれぞれ概略つぎのように動作する。

【００２５】

計算機１１～１ｎの各々は同一の機能を持つので、以下では１ｉで代表させる

。計算機１ｉ上の負荷データ測定手段Ａ１ｉ１は、一定時間ごとに自身の属する計算機１ｉの負荷データを測定し、結果を中継配分装置２に送る。トランザクション処理手段１ｉ２は、中継配分装置２から送られた処理要求メッセージを通信手段１ｉ３から受け取ると、自身の管理下の業務処理プロセスを割り当て、処理を行わせる。前記業務処理プロセスは、プログラムの実行のためにＣＰＵの使用とファイル装置４上のファイルへのアクセスを繰り返し、処理が終了すると応答メッセージを作成し、通信手段１ｉ３を介して中継配分装置２に送る。

【００２６】

中継配分装置２上の負荷データ記憶手段６には各計算機の負荷データが格納されている。負荷データ測定手段Ｂ８は、各計算機から一定時間ごとに送られてくる負荷データを通信手段２１経由で受け、これを加工して推定データとして前記負荷データ記憶手段６に格納する。また、計算機へのトランザクション処理要求送付および応答メッセージ到着の通知を通信手段２１から受け、負荷データ記憶手段６上の一部のデータを更新する。実行計算機選択手段７は、前記処理要求メッセージを通信手段２１から渡され、負荷データ記憶手段６に記憶されている各計算機の前記推定負荷データから各計算機における推定伸長率を求め、これに基づいて実行すべき計算機を決定し、該計算機に向けて、通信手段２１に処理要求メッセージを送付させる。

【００２７】

次に、図１～図５を参照して本実施の形態の全体の動作について詳細に説明する。

【００２８】

図１において、トランザクション処理手段１ｉ２は、中継配分装置２から送られた処理要求メッセージを通信手段１ｉ３から受け取ると、自身の管理下の業務処理プロセスを割り当て、要求に応じたトランザクション処理を行わせる。トランザクション処理手段１ｉ２は、複数のトランザクション、したがって複数の業務処理プロセスをマルチプログラミング状態で走らせることができ、これによって応答時間、資源使用効率を向上させている。業務処理プロセスは、適用業務プログラム実行のためにＣＰＵの使用とファイルアクセスのためのファイル装置群

４への入出力を繰り返し、処理が終了すると応答メッセージを作成し、通信手段１ｉ３を介して中継配分装置２に送りプロセスを終了する。

【００２９】

図２に、１台の計算機ｉについて、性能面から見たシステムのモデルを示す。業務処理プロセスの資源使用特性はＣＰＵ使用時間と入出力の回数で捉えられるが、これはトランザクションごとに異なるものである。複数の処理を並行して走らせるので、資源の競合が起こる。そのため、少なくともＣＰＵの前にはプロセス待ち行列ができることが想定される。一般に、使用率の高いＣＰＵほど待ち時間が長い。入出力に関しては、いずれの計算機からも性能的に同条件にあるので、ここの処理時間は待ち時間も含めてアクセス元の計算機による差はないと考える。Ｎｉは計算機ｉ上の処理中業務処理プロセス数を表すものとする。これは該計算機上で処理中のトランザクション数に相当する。ＰｉはＣＰＵ系に存在する業務処理プロセス数を表すものとする。これはＮｉのうちＣＰＵ割当て待ち（レディ状態）あるいはＣＰＵ使用中であるプロセスの総数である。ファイル装置は共有されており、すべての計算機についてアクセス性能は同等なので、他の計算機も、性能的にはすべて計算機ｉと同じ位置づけとなる。

【００３０】

計算機１ｉ上の負荷データ測定手段Ａ１ｉ１は、一定時間ごとに自身の属する計算機１ｉの負荷データとして、その時点でＣＰＵ系に存在する前記業務処理プロセス数Ｐｉあるいは直前の測定以後今回までの間のＣＰＵ使用率Ｒｉ、およびその時点での処理中業務処理プロセス数Ｎｉを測定し、結果を通信手段１ｉ３を介して中継配分装置２に送る。前記測定の間隔は負荷分散の精度を左右するので、オーバーヘッドとの兼ね合いもあるが、通常のトランザクション処理では１００ミリ秒程度以下、できれば１０ミリ秒程度、であることが望ましい。Ｐｉを用いるかＲｉを用いるかは実施システムごとに決定してよい。Ｐｉを用いる場合をＰ方式、Ｒｉを用いる場合をＲ方式と呼ぶことにする。

【００３１】

図３に、中継配分装置２上の負荷データ記憶手段６に記憶する負荷データをテーブル形式で示す。計算機番号Ｔ１はシステム内で稼働中の計算機の識別を示し

、テーブル上のデータは計算機ごとに１行を用意して管理されている。負荷データ測定手段Ｂ８は、前記負荷データ測定手段Ａ１ｉ１から一定時間ごとに送られるデータである、ＣＰＵ系に存在する業務処理プロセス数Ｐｉあるいは直前の測定以後今回までの間のＣＰＵ使用率Ｒｉ、およびその時点での処理中業務処理プロセス数Ｎｉの値を通信手段２１経由で受ける。そして、これを加工して推定データとして、前記Ｐｉの推定値Ｐｅｉあるいは前記Ｒｉの推定値Ｒｅｉ、および前記Ｎｉの推定値Ｎｅｉを求め、前記Ｒ方式ならＲｅｉからＰｅｉを計算し、列Ｔ３にＮｅｉを列Ｔ４にＰｅｉをそれぞれｉ番目の値として格納する。

【００３２】

測定値をそのまま用いず推定値に変換するのは、過去のデータを総合的に組み込むことによりサンプリングの信頼性の低さを補うためである。具体的な求め方として、次の方法がある。測定値をｍ、推定値をｅ、最新の測定がｎ回目であったとする。

ｅ（ｎ）＝ａ＊ｍ（ｎ）＋（１－ａ）＊ｅ（ｎ－１）　（式３）

ここで、ａはパラメタ（０＜ａ≦１）であり、また、ｅ（ｎ）の初期値ｅ（０）はｅ（１）と等しいとする。すなわち、今回の測定値にａを乗じたものと前回の推定値に１－ａを乗じたものとの和を今回の推定値とする。式３は次のように展開できる。

ｅ（ｎ）＝ａ＊ｍ（ｎ）＋ａ（１－ａ）＊ｍ（ｎ－１）＋ａ（１－ａ）$^{2}$＊ｍ（ｎ－２）＋ａ（１－ａ）$^{3}$＊ｍ（ｎ－３）＋…

この式は、推定値が、近い過去の測定値ほど重視する形で過去の測定値を全部取り込んだものになっていることを示している。ａが大きい（１に近い）ほど近い過去を重視する度合いが高いことになる。前記測定間隔が十分に小さいなら、ａの値は０．１などの小さい値とした方が推定値の信頼度は上がる。前記Ｒ方式では、式３により求められたＲｅｉから次の式によりＰｅｉを求め、列Ｔ４に格納する。

Ｐｅｉ＝Ｒｅｉ／（１．０－Ｒｅｉ）（Ｒｅｉ≧０．９９ならＰｅｉ＝Ｎｅｉ）

この式は、Ｍ／Ｍ／１待ち行列における系の長さと使用率との関係そのものである。

【００３３】

また、計算機ｉ上で処理中のトランザクション現在数Ｎｐｉが、テーブル上の列Ｔ２に保持されている。この値は、前記負荷データ測定手段Ｂ８が、計算機へのトランザクション処理要求送付（トランザクション開始）および応答メッセージ到着（トランザクション終了）の通知を通信手段２１から受けて更新し、保持する。したがって、信頼できる測定値である。

【００３４】

図４Ａは、本発明の第１の実施の形態の実行計算機選択手段７の動作を示すフローチャートである。実行計算機選択手段７は、端末から処理要求メッセージが中継配分装置２に到着すると通信手段２１に起動されて該メッセージを渡され（図４Ａの７１）、負荷データ記憶手段６に記憶されている各計算機の負荷データから各計算機における推定伸長率を求めて（図４Ａの７２，７３）負荷指標の値を計算し（図４Ａの７４）、これ基づいて実行すべき計算機を決定し（図４Ａの７５）、該計算機に向けて、通信手段２１に処理要求メッセージを送付させる（図４Ａの７６）。

【００３５】

推定値の補正（図４Ａの７２）では、負荷データ記憶手段６上のデータをベースに補正を行い、現時点における負荷データとして、処理中業務処理プロセス数（補正値）Ｎｒｉ、およびＣＰＵ系に存在する業務処理プロセス数（補正値）Ｐｒｉを次の式により求める。

Ｎｒｉ＝ｗ＊Ｎｐｉ＋（１－ｗ）＊Ｎｅｉ　（式４）

ここでｗは重み係数（ｗ≦１．０）であり、０．８程度がよい。

Ｐｒｉ＝Ｐｅｉ＋（Ｎｒｉ－Ｎｅｉ），Ｎｒｉ≧Ｎｅｉのとき　（式５）

　　＝Ｐｅｉ＊（Ｎｒｉ／Ｎｅｉ），Ｎｒｉ＜Ｎｅｉのとき

式５は、過去のサンプリングから推定していたＰｅｉを、これと同一条件で推定したＮｅｉと現時点の状況を表す最も信頼できる補正値であるＮｒｉとの関係から、補正するものである。一定時間間隔でしか行わない推定をベースに、最新の推定値から現時点の真の値に近い補正値を得ることができる。

【００３６】

推定伸長率の計算（図４Ａの７３）では、推定伸長率Ｅｐｉを次の式により求め、結果をテーブルの列Ｔ５に格納する。

Ｘ＝Ｎｒｉ＊（Ｐｒｉ＋１）として、

Ｅｐｉ＝Ｘ／（Ｘ－Ｐｒｉ＊Ｐｒｉ），Ｐｒｉ＜Ｎｒｉのとき　（式６）

　　＝Ｎｒｉ＋１．０　　，Ｐｒｉ≧Ｎｒｉのとき

【００３７】

式６は、図２に示す１つの計算機（ＣＰＵ１台）における平衡状態の平均値に関して成立する関係から、以下のようにして導かれる。

【００３８】

トランザクションは指数分布に従う時間間隔で到着する（ポアソン到着）とする。また、ディスク装置では待ちは生じない（装置が無限に存在する）ものとする。前述の処理中業務プロセス数、その内のＣＰＵ系に存在する業務処理プロセス数、ＣＰＵ使用率もここでは平均値とし、これらを含めてすべての変数は計算機番号、推定状態を示す添字を省いて示す（例えば、処理中業務プロセス数はＮ、ＣＰＵ系に存在する業務処理プロセス数はＰ、ＣＰＵ使用率はＲでそれぞれ示す）。また、以下で定義する４種の変数はトランザクション当たりの平均時間とする。

Ｆ：処理時間

ｔ：純処理時間

ｓ：ＣＰＵ使用時間

ｄ：入出力時間（ｔ＝ｓ＋ｄ）

更に、対象計算機からの要求で実行中の入出力数の平均をＤとする（Ｎ＝Ｐ＋Ｄ）。

従来技術４の文献２２８頁の８．３式から、平衡状態の平均値について、

Ｆ＝ｓ（Ｐ＋１）＋ｄ＝ｓＰ＋ｔ　　（式６１）

となり、入出力で待ちがないので、

ｄ／ｓ＝Ｄ／Ｒ　（式６２）

となる。また、同文献２２８頁の式８．３と式８．１の対比からも知られるように、

Ｐ＋１＝１／（１－Ｒ）　から　Ｒ＝Ｐ／（１＋Ｐ）　式（６３）

となる。

式６３を式６２に代入し、ｄ＝ｔ－ｓ、Ｎ＝Ｐ＋Ｄを適用すると、

ｓ＝Ｐｔ／（Ｐ＋Ｄ＋ＰＤ）＝Ｐｔ／（Ｎ（Ｐ＋１）－Ｐ・Ｐ）　式（６４）

となり、式６４を式６１に代入すると、

Ｆ＝Ｐ・Ｐｔ／（Ｎ（Ｐ＋１）－Ｐ・Ｐ）＋ｔ

＝Ｎ（Ｐ＋１）ｔ／（Ｎ（Ｐ＋１）－Ｐ・Ｐ）

＝Ｘｔ／（Ｘ－Ｐ・Ｐ）

となる。ここで、Ｘ＝Ｎ（Ｐ＋１）である。

したがって、伸長率Ｅは、

Ｅ＝Ｆ／ｔ＝Ｘ／（Ｘ－Ｐ・Ｐ）

となり、式６が導かれる。

また、伸長率Ｅは、ＮとＣＰＵ使用率Ｒを用いて次のように表すこともできる。

Ｅ＝Ｎ（１－Ｒ）／（Ｎ（１－Ｒ）－Ｒ・Ｒ）　式（６５）

式６５は、式６にＲとＰの関係を表す式６３を適用して得ることもできるし、従来技術４の文献の式８．１（Ｒを用いて処理時間を表現）から出発して、式６２、式６３を適用して得ることもできる（導出の記述は省略する）。しかし、Ｒについては式５で行ったのに相当する補正の手段がなさそうなので、本実施形態では補正の前にＰに変換してから補正を受けるようにしてしまい、ＣＰＵ使用率を測定した場合にも最終的な伸長率の式としてはＮとＰを用いる式６を使用するようにした。

なお、本実施形態で推定伸長率を求めるために用いる計算式（式６）は、このようにシステムの統計的平衡状態に関して成立するものであり、平衡状態がある程度の時間続くときに、その間のＰ、Ｎの平均値を知れば推定可能になるものである。現状を表す平衡平均値としては、過去の履歴に基づいて式３を用いたＮｅｉ、Ｐｅｉが適当と考えられるが、Ｎについては正確な現在値Ｎｐｉが知られているので、動的負荷配分の立場からはこれも反映すべく、前述した式４においては、この方針によりＮの補正値を得ている。

【００３９】

再び図４（Ａ）を参照して実行計算機選択手段７の残りの動作を説明する。

【００４０】

採用する負荷指標によっては、現状における推定伸長率Ｅｐｉの他に到着メッセージを計算機ｉにスケジュールした場合の予測伸長率Ｅｎｉが必要になる。あるいはＥｎｉだけを必要とすることもある。ＥｎｉもＥｐｉと同様に図４（Ａ）のステップ７３で計算される。

【００４１】

Ｅｎｉが必要であって到着メッセージ処理のジョブ特性を利用しない場合は、スケジュール後のトランザクション数ＮｎｉをＮｒｉ＋ｗとし、スケジュール後のＣＰＵ系滞在プロセス数ＰｎｉをＰｒｉ＋ｗとして、式６と同様にＥｎｉを計算しテーブルの列Ｔ６に格納する。

【００４２】

到着メッセージ処理のジョブ特性が推定可能でこれを利用する場合は、メッセージの種類などからその純処理時間（ＣＰＵ、ファイル装置という資源を実際に使用する時間の合計、言い換えると資源競合が全くない場合の処理時間）に占めるＣＰＵ時間の割合Ｃを推定し、これを用いて次の計算により、まず前記Ｐｎｉを推定する。

【００４３】

$C_1$をスケジュール前における計算機ｉ上におけるＣの推定値とすると、$C_1$は式６４におけるｓ／ｔに相当するので、式６４にＮ、Ｐの補正値を当てはめて、

$C_1 = Pri / (Nri * (1 + Pri) - Pri \cdot Pri)$

となる。$C_2$を計算機ｉに、$C = C_0$である到着メッセージをスケジュールした場合の新ジョブミックスにおけるＣの推定値とする。平均が$C_1$のジョブがＮｒｉ個存在し、そこへ$C_0$のものが１個加わり総数はＮｎｉとなるので、その平均値は、

$C_2 = (Nri * C_1 + C_0) / Nni$

となる。

式６４から、ｓ／ｔ＝Ｐ／（ＮＰ＋Ｎ－Ｐ・Ｐ）なので、この式をスケジュール後の状態に適用すると、ｓ／ｔ＝$C_2$なので、Ｐｎｉをｙとおくと、

$C_2$（Ｎｎｉ・ｙ＋Ｎｎｉ－$y^2$）＝ｙ

となり、整理すると、次の２次方程式が得られる。

$C_2 y^2 + (1 - C_2 \cdot Nni)\, y - C_2 \cdot Nni = 0$　　（式７）

式７をｙについて解くことによって、スケジュール後のＰｎｉの推定値が得られる。定数項が負の値なので正の解と負の解が得られる。正の解をＰｎｉとして採用する。そして、式６と同様にしてＥｎｉを計算する（これを、以下ではＥｋｉとする）。

【００４４】

ここで処理時間の伸長率とは、業務処理プロセスの応答時間、すなわち待ち時間も含む処理時間の、純処理時間に対する倍率を表す。伸長率Ｅｉは、計算機ｉにおける業務処理プロセスの伸長率である。処理速度が同じ計算機ならば、同一の処理は伸長率の小さい計算機で実行した方が処理時間は短く、したがって応答時間を短くできることになる。推定伸長率は、当該計算機上で実行中のプロセスの集まり（ジョブミックス）の、動作中の群としてのプログラム特性（ＣＰＵ使用特性だけでなく、ＣＰＵ－Ｉ／Ｏ使用特性を含む）を反映している。しかも、式６を用いると、実行中の個々のジョブの特性を知る必要がなく、動作中に観測可能なデータだけから得ることが可能なところに特徴がある。基本的に、従来技術４の考え方の系列に属し、式６は式２の拡張・変形により得られるが、当方式は現時点のシステム状況（ジョブミックス特性）を反映可能にし、かつ、ＣＰＵ系での滞在時間だけでなく入出力も含めた全処理時間（応答時間）を対象にして、精度・ダイナミック性を向上させている。ただし、式６は平衡平均値に関する理論に基づいているので、短期的な状況の把握法としては１００パーセントの信頼性があるとは言えない。

【００４５】

負荷指標の値の計算（図４Ａの７４）では、負荷データＴ２～Ｔ６を用いて各計算機について負荷指標の値を計算する。負荷指標としては図５に示すように８種類（名称としてＬで始まる）が考えられ、実施システムではこの内の一種類を

選べばよい。図中に示した式による計算で結果を得てテーブルの推定負荷の列Ｔ７に格納する。いずれも、小さい値を持つ計算機ほどスケジュール先として望ましいことになる。どの時点の負荷を考えるかについて、到着メッセージのスケジュール前（この負荷をＬｐと表記する）／後があり、さらに、スケジュール後の場合に到着メッセージのジョブ特性を未知とする（負荷をＬａと表記）か、推定可能とする（負荷をＬｋと表記）かがありうる。これら３ケース各々について、伸長率そのものを負荷指標と捉える（Ｌｘ１と表記）こともでき、推定伸長率に処理中業務処理プロセス数ＮｒｉまたはＮｎｉを乗じたものを負荷指標とする（Ｌｘ２と表記）こともできる（ｘはｐ、ａまたはｋである）。後者は計算機上の個々のトランザクションの推定伸長率の総和という性格をもつ。さらに、スケジュールによる負荷の増加という観点から、上記の総和のスケジュール前後における増分を負荷指標とする（Ｌｘ３と表記）こともできる。伸長率の増分最小という選択は、システム全体にとって当スケジュールによる応答時間総和の増加を最小にする選択になり、結果として平均応答時間を最小化できると期待できる。到着メッセージのジョブ特性が相当の精度で推定可能な場合は、理論通り、Ｌｋ３を採用するのが最も良い結果を期待できる。ジョブ特性推定の精度が期待できない場合はＬｋ３の選択は危険であり、平均応答時間最小という点からはＬａ２を採用するのがよい。

【００４６】

推定値の補正（図４Ａの７２）、推定伸長率の計算（７３）、負荷指標の値の計算（７４）は、入力メッセージを処理可能なすべての計算機に関して行い、推定負荷を得てテーブルの列Ｔ７に格納しておく。

【００４７】

実行すべき計算機の決定（図４Ａの７５）では、テーブルの列Ｔ７に格納されている各計算機の前記推定負荷をサーチし、推定負荷が最小の計算機（計算機ｊとする）を選択する。次に、メッセージの送付（図４Ａの７６）では、選択された計算機ｊに対して入力メッセージを送付して処理開始を促すように、通信手段２１に指令する。

【００４８】

次に、本実施の形態の効果について説明する。

【００４９】

本実施の形態では、中継配分装置２の上で全計算機の負荷データをリアルタイムで管理し、またすべての処理要求メッセージを直接受け取り、直ちに、前記負荷データに基づいて各計算機における伸長率を計算し、その時点で最適な負荷指標値をもつ計算機に処理要求メッセージの処理を依頼するように構成されているため、集中的な制御が実現でき、オーバヘッドの少ない、かつ、良質な負荷配分を実現することができる。

【００５０】

次に、本発明の第２の実施の形態について図面を参照して詳細に説明する。

【００５１】

図６を参照すると、本発明の第２の実施の形態は、第１の実施の形態に対し、構成として、中継配分装置２をもたず端末群５は通信網５１を介して直接に計算機群１に接続されている点と、計算機間を接続する交換・蓄積機構１０が追加されている点が異なる。これに伴い、計算機１ｉは、負荷データ測定手段Ａ１ｉ１と、トランザクション処理手段１ｉ２と、通信手段１ｉ３に加えて、負荷データ記憶手段６ｉと、実行計算機選択手段７ｉと、負荷データ測定手段Ｂ８ｉとを含む。これらの各手段を他の手段と共にソフトウェア的に実現する場合、第１の実施の形態と同様にその実現用プログラムが図示しない記録媒体に記録されて提供される。

【００５２】

１つのトランザクションの処理の概略は次のようになる。トランザクション処理要求であるメッセージは端末群５に属する端末装置から送り先を指定して送り出され、通信網５１を経由して指定された計算機で受け取られる。受け取った計算機はそのメッセージを自分で処理するか他に依頼するか、依頼するとしたらどの計算機にするかを決定し、依頼する場合は交換・蓄積機構１０を介して依頼先計算機に該メッセージを送る。処理を行う計算機はトランザクション処理を実行し、応答メッセージを作成して要求元端末に返す。

【００５３】

ここで、上記の手段はそれぞれ概略つぎのように動作する。

【００５４】

計算機１ｉ上の負荷データ測定手段Ａ１ｉ１は、一定時間ごとに自身の属する計算機１ｉの負荷データを測定し、負荷データ測定手段Ｂ８ｉでこれを加工して推定データとして負荷データ記憶手段６ｉに格納する、と共に交換・蓄積機構１０により他のすべての計算機に通知する。また、同様に各計算機のトランザクション処理開始および終了を相互に通知し合う。これらによって、各計算機上の負荷データ記憶手段６ｘには全計算機の最新の負荷データが保持される。端末から来た処理要求メッセージは実行計算機選択手段７ｉが受け、負荷データ記憶手段６ｉに記憶されている各計算機の推定負荷データから各計算機における推定伸長率を求め、これに基づいて実行すべき計算機を決定し、自身で実行する場合にはトランザクション処理手段１ｉ２に渡し、他の計算機に実行させる場合は該計算機のトランザクション処理手段１ｊ２に向けて、交換・蓄積機構１０を経由して処理要求メッセージを送付する。トランザクション処理手段１ｘ２は、処理要求メッセージを受け取ると、自身の管理下の業務処理プロセスを割り当てて処理を行わせる。業務処理プロセスは、プログラムの実行のためにＣＰＵの使用とファイル装置４上のファイルへのアクセスを繰り返し、処理が終了すると応答メッセージを作成し、通信手段１ｘ３を介して要求元端末に送る。

【００５５】

次に、図６及び図４Ｂのフローチャートを参照して本実施の形態の全体の動作について詳細に説明する。

【００５６】

トランザクション処理手段１ｉ２の動作は、処理要求メッセージを受け取るのが自分または他の計算機上の実行計算機選択手段７ｘからである点を除くと第１の実施の形態と同一である。負荷データとして管理するデータも第１の実施の形態と同一で図３に示すものであるが、各計算機上に負荷データ記憶手段６ｘとして、全計算機に関する同一内容のものを保持する。計算機１ｉ上の負荷データ測定手段Ａ１ｉ１は、一定時間ごとに自身の属する計算機１ｉの負荷データとして、ＣＰＵ系に滞在する業務処理プロセス数ＰｉあるいはＣＰＵ使用率Ｒｉ、およ

びその時点での処理中業務処理プロセス数Ｎｉを測定する。そして、負荷データ測定手段Ｂ８ｉがこれを加工して、推定データとして、ＰｅｉあるいはＲｅｉ、およびＮｅｉを求め、前記Ｒ方式ならＲｅｉからＰｅｉを計算し、列Ｔ３にＮｅｉを列Ｔ４にＰｅｉをそれぞれｉ番目の値として格納する。同時にＮｅｉ、Ｐｅｉの値を他のすべての計算機に交換・蓄積機構１０を介して送り、負荷データ記憶手段６ｘの内容を更新させる。推定データの計算方法は第１の実施の形態におけるのと同一である。また、計算機１ｉ上で処理中のトランザクション現在数Ｎｐｉ（テーブル上の列Ｔ２）に関しては、負荷データ測定手段Ｂ８ｉが、計算機１ｉでのトランザクション処理開始およびトランザクション処理終了の通知をトランザクション処理手段１ｉ２から受けて更新・保持する、と共に他のすべての計算機に送り負荷データ記憶手段６ｘの内容を更新させる。

【００５７】

図４Ｂは、本発明の第２の実施の形態の実行計算機選択手段７の動作を示すフローチャートである。端末から処理要求メッセージが到着した計算機１ｉにおいて実行計算機選択手段７ｉが実行される。到着メッセージを受けた通信手段１ｉ３に起動されて該メッセージを渡され（図４Ｂの７１）、負荷データ記憶手段６ｉに記憶されている各計算機の負荷データから各計算機における推定伸長率を求めて（図４Ｂの７２，７３）負荷指標の値を計算し（図４Ｂの７４）、これ基づいて自計算機（１ｉ）で実行すべきかどうか判断し（図４Ｂの７５１）、自計算機ですべきでないなら実行する計算機を決定し（図４Ｂの７５）、選択された計算機に向けて、処理要求メッセージを交換・蓄積機構１０を介して送付させる（図４Ｂの７６）。自計算機で実行すべきなら自分を選択し（図４Ｂの７５２）、処理を指示する（７６）。実行計算機選択手段７ｘの動作として第１の実施の形態と論理的に異なるのは、自計算機で実行するか否かの判断のところだけである。　推定値の補正（図４Ｂの７２）、推定伸長率の計算（７３）、負荷指標の値の計算（７４）は、入力メッセージを処理可能なすべての計算機に関して行い、推定負荷を得てテーブルの列Ｔ７に格納しておく。自計算機で実行するか否かの判断（７５１）には、まず、自計算機の現推定伸長率Ｅｐｉを用い、これが閾値（小さめに１．３程度がよい）以下であったならば、自計算機で実行することに

する。そうでない場合、負荷データ記憶手段６ｉ上の推定負荷Ｔ７に基づいて、自計算機の推定負荷が最小でなくても、最小負荷の計算機との差が小さければ自計算機で実行するようにする。較差の大小の判断は閾値による。負荷指標として伸長率総和の増分以外の６種類のいずれかを採用する場合は、較差の判断は倍率閾値によるのがよく、１．３倍から１．５倍程度がよいようである。すなわち、自計算機のＥｐｉがＥｐが最小である計算機ｊのＥｐｊの１．３倍以内であったら自計算機で実行する、などである。負荷指標として伸長率総和の増分を採用する場合は、２台の計算機の負荷指標間の差をシステム内の全トランザクション数で除したものが閾値を越えるか否かで判断するのがよい。すなわち、実行中の全トランザクションの平均伸長率の増分の程度によって判断する。この場合の閾値は０．０２程度がよい。ここで、閾値の値は、負荷の測定間隔が長い場合には大きくした方がよい。これは、測定間隔が長い場合は負荷指標の推定値の信頼性が低くなるので、トランザクション転送を行う頻度が少なくなるような安全サイドの選択をした方がいいからである。到着計算機で処理を実行してしまうことを優先するのは、他の計算機に転送するには転送元・転送先の双方にオーバヘッドがかかり、また、対象トランザクション自身の処理時間に遅延をもたらすからである。閾値を用いた判断を入れないと、ほとんどすべてのメッセージのトランザクション処理を他の計算機へ依頼する結果になる可能性が高い。

【００５８】

自計算機で処理することに決定したらトランザクション処理手段１ｉ２にメッセージを引き渡し処理を依頼する。　到着計算機で処理すべきでないとなったときは、実行すべき計算機の決定（図４Ｂの７５）で、テーブルの列Ｔ７に格納されている各計算機の推定負荷をサーチし、推定負荷が最小の計算機（計算機ｊとする）を選択する。そして、メッセージの送付（図４Ｂの７６）で、選択された計算機ｊのトランザクション処理手段１ｊ２に対して、入力メッセージを交換・蓄積機構１０を介して送付し、処理開始を促す。

【００５９】

以上において、負荷データ記憶手段６は同一内容のものが各計算機上に保持されるとし、各計算機で自身に関して測定／計算後に他のすべての計算機に交換・

蓄積機構１０を介して通知するとしていたが、交換・蓄積機構１０がある程度の容量をもち主記憶程度に速い蓄積機構を備えるなら、前記負荷データ記憶手段６の一部は、システム共用のものとして交換・蓄積機構１０の上に格納し保持することもできる。負荷データ更新のオーバヘッドの観点から、この構成の方が望ましい。この場合、テーブルの列Ｔ１～Ｔ４は交換・蓄積機構１０に保持し、各計算機はメッセージが到着した際に、ここから引き出したデータに基づいて推定伸長率、推定負荷などを計算し、処理を実行すべき計算機を決定すればよい。また、交換・蓄積機構１０が前述の条件を満たす場合、他の計算機に処理を依頼することになったときには、メッセージそのものを直接送付するのでなく、メッセージは交換・蓄積機構に格納し、依頼の通知だけを相手に送るように構成することもできる。この場合、受け取り側の計算機は、都合の良いときに非同期的に交換・蓄積機構から取り出すことになる。

【００６０】

次に、本実施の形態の効果について説明する。

【００６１】

本実施の形態では、特別な中継配分装置を備えなくてもよいため、システム全体として低コストで構成することができる。集中制御による負荷分散はできないが、処理要求メッセージは端末からの指定により送付された先の計算機で、その計算機及び他の計算機の負荷状況データに基づいて伸長率を計算し、転送のオーバヘッドも考慮した上で、その時点で最適な実行計算機を決定し、その計算機上で実行させるように構成されているため、集中制御である第１の実施の形態よりは落ちるが、分散制御下としては高い応答性能を実現できる。

【００６２】

図８及び図９に示すグラフは、本実施の形態におけるような分散制御下における負荷分散の効果を、シミュレーション評価によって確認した結果である。トランザクションとしては、純処理時間（４５０ミリ秒）に占めるＣＰＵ時間の割合が平均５％のものと平均６０％のものの２種類が、７対３の割合で到着するとした。計算機は８台あり、各計算機への到着はランダムで、平均としては等しい到着率になるように設定した。横軸は到着率に比例する負荷率を示し、縦軸は図８

では得られた平均応答時間（ミリ秒）であり、図９では応答時間のばらつき（標準偏差）である。それぞれのグラフ曲線は負荷分散方式に対応しており、実線のものが本実施の形態に関係する。ＮＣ方式は負荷分散をせず、到着したものをそのまま処理する。ＭＰＬ方式は、処理中トランザクション現在数を負荷指標とする動的制御で、これが到着計算機より２以上小さい計算機が存在したら、最小の計算機に転送し処理させる。Ｌａ２、Ｌｋ３はそれぞれ本実施の形態における推定伸長率に基づく負荷指標を用いた動的制御に対応する。これらの結果から、平均応答時間について、静的確率的配分としては最適であるはずのＮＣ方式よりも動的制御は大幅によいことが分かり、特に推定伸長率に基づく方式は従来多く用いられている実行中トランザクション数に基づく方式よりも優れていることが示され、また、応答時間のばらつきについても同様な傾向が、より顕著に現れていることが分かる。このような差は負荷率が高いときに、より顕著である。

【００６３】

次に、本発明の第３の実施の形態について図面を参照して詳細に説明する。

【００６４】

図７を参照すると、本発明の第３の実施の形態は、第２の実施の形態に対し、構成として、中継仮配分装置２５をもち、端末群５は中継仮配分装置２５を経由して計算機１１～１ｎに接続されている点だけが異なる。

【００６５】

１つのトランザクションの処理の概略は次のようになる。トランザクション処理要求であるメッセージは端末群５に属する端末装置から送り出され、前記中継仮配分装置２５に渡される。中継仮配分装置２５は受け取ったメッセージを処理する計算機１ｉを仮決定し、該計算機に該メッセージを送る。受け取った計算機１ｉはそのメッセージを自分で処理するか他に依頼するか、依頼するとしたらどの計算機にするかを決定し、依頼する場合は交換・蓄積機構１０を介して依頼先計算機に該メッセージを送る。処理する計算機はトランザクション処理を実行し、応答メッセージを作成して要求元端末に返す。ここで、第２の実施の形態に対して追加された中継仮配分装置２５は、端末からのメッセージを受けて仮配分先計算機を決定して送付するが、基本的に、配分は詳細な動的情報に基づかない、

静的／準静的な手法によって行われる。

【００６６】

次に、本実施の形態の全体の動作について詳細に説明する。

【００６７】

中継仮配分装置２５は、第１の実施の形態における前記中継配分装置２と同様に、端末装置群５から送り出されるすべての処理要求メッセージを受け取り、これを渡すべき計算機を決定して送付する。中継仮配分装置２５における静的／準静的な仮配分方式として次の３種類が想定される。実施に当たっては、このうちいずれか１種類か、あるいはこれらを組み合わせた方式を選択する。これら以外であっても、計算機からの負荷データの収集が少なく、実行のオーバヘッドも小さい配分方式なら採用可能である。

（１）端末のグループ分けによる配分

（２）到着順に、巡回的に各計算機へ配分

（３）実績データに基づく確率的配分

【００６８】

（１）端末グループ分けによる配分では、メッセージ発生元の端末によって配分先の計算機を固定的に定めておく。すなわち、端末群を計算機１で処理するグループ、計算機２で処理するグループ、のように予めグループ分けしておき、どの端末から来たかによって行く先を機械的に決定する。第２の実施の形態とほとんど同じ方式になるが、本方式では端末群と計算機との対応関係を中継仮配分装置で集中的に管理できるので、過去の実績に応じて、例えばシステム立ち上げの度ごとに、長期的には負荷バランスのとれるグループ分けに設定し直すなどを容易にできる。

【００６９】

（２）到着順に巡回的に各計算機へ配分では、中継仮配分装置に到着した最初のメッセージは計算機１へ、次は計算機２へ、と順次配分し、最後の計算機ｎに配分した次のメッセージは再び計算機１へ、と巡回的に配分する。特に大部分のメッセージ処理のジョブ特性が同一クラスに属するような場合、短期的にも負荷をバランスさせる効果が期待できる。

【００７０】

（３）実績データに基づく確率的配分では、各計算機に配分するメッセージ数の比率を計算機ごとに設定し、短期的にもこの比率を守るように配分をする。各計算機から負荷状況のデータを１秒ごと、１０秒ごとなどに定期的に受け取り、負荷がアンバランスであったなら、バランスさせるように個々の計算機への配分比率を上下させ、以後はこの配分比率に基づいて配分を行うようにする。

【００７１】

図７におけるトランザクション処理手段１ｘ２の動作、各計算機上にある負荷データ記憶手段６ｘの内容は第２の実施の形態と同一である。計算機１ｘ上の負荷データ測定手段Ａ１ｘ１、負荷データ測定手段Ｂ８ｘも第２の実施の形態におけるのと同一の動作をするが、それに加えて負荷データ測定手段Ｂは、中継仮配分装置２５が前述の配分方式（３）を採用する場合、１秒、１０秒などの間隔で負荷データの概要を中継仮配分装置２５に送る。本発明の第３の実施の形態の実行計算機選択手段７の動作は、第２の実施の形態におけるのと同一であり、図４Ｂのフローチャートで示される。

【００７２】

以上において、負荷データ記憶手段６は同一内容のものが各計算機上に保持されるとし、各計算機で自身に関して測定／計算後に他のすべての計算機に交換・蓄積機構１０を介して通知するとしていたが、負荷データ記憶手段６はシステム共用のものとして交換・蓄積機構１０の上に格納し保持することもできる。負荷データ更新のオーバヘッドの観点から、この構成の方が望ましい。また、他の計算機に処理を依頼することになったときには、処理対象メッセージそのものを直接送付するのでなく、メッセージは交換・蓄積機構に格納し、依頼の通知だけを相手に送るように構成することもできる。以上の点に関しても、第２の実施の形態におけるのと同様である。

【００７３】

次に、本実施の形態の効果について説明する。

【００７４】

本実施の形態では、中継配分装置として、限定された機能だけをもち、計算機

群１１～１ｎからの情報収集量・頻度も小さいものを備えるだけでよいので、比較的低コストで全体システムを構成できる。機能の限定された中継仮配分装置であるが、ここで準静的にとはいえ負荷の適切な仮配分を行うように構成されているため、中継仮配分装置が存在しない場合と比較して、応答性能（平均、ばらつき共）を向上させることができ、また、計算機に到達してから行われる負荷バランスのための転送の頻度を大幅に減少させることができる。

【００７５】

図８及び図９は、第２の実施の形態の効果の説明で前述した条件の下で、シミュレーション評価により得られたものであり、第３の実施の形態の結果が点線のグラフとして含まれている。グラフ曲線はそれぞれ負荷分散方式に対応している。Ｒ＿ＮＣは、仮配分として前記（２）到着順に巡回的に配分を実施して、仮配分先の計算機でそのまま処理を実行させたものである。Ｒ＿Ｌｋ３は、仮配分を同じく（２）で行い、仮配分先の計算機で前記Ｌｋ３を負荷指標とする負荷配分を行った結果である。これから、準静的な配分だけでも静的な配分であるＮＣ方式よりも応答性が大幅に向上することが分かり、さらに推定伸長率に基づく負荷分散を組み合わせることにより、第２の実施の形態によるよりも応答性を向上させられることが理解できる。

【００７６】

【発明の効果】

本発明によれば、複数の計算機が負荷分担してトランザクション処理を実行するシステムにおいて、短期的にも計算機間の負荷をバランスさせ、全体として応答時間の平均及びばらつきを小さく保つことができる。その理由は、計算機の処理時間の伸長率をベースとする新規な負荷指標に基づいて処理要求を動的に配分しているためである。つまり、基本的な性能指標として処理時間の伸長率は、当該計算機上で実行中のプロセスの集まり（ジョブミックス）の、動作中の群としてのプログラム特性（ＣＰＵ特性だけでなく、ＣＰＵ－Ｉ／Ｏ使用特性を含む）を反映しており、対話型処理にとっては最適で、かつ、適用性が広いためである。

【００７７】

また本発明によれば、各計算機における処理時間の伸長率を、一定時間ごとの各計算機の処理中トランザクション数、ＣＰＵ系に滞在する業務処理プロセス数、ＣＰＵ使用率といった、動作中に観測可能なデータから導き出せるようにしたため、個々の処理要求のジョブ特性に関する先験的知識なしに、低オーバヘッドで実測可能な負荷データだけに基づいて、伸長率をベースとした負荷指標に基づく負荷分散が実現できる。

【００７８】

また一定時間ごとに測定した負荷データの系列を総合的に用いて各計算機の負荷状況を推定し、あるいは各計算機におけるトランザクション処理の開始・終了に応じて各計算機の処理中トランザクション現在数を常に把握しておき、この処理中トランザクション現在数を用いて推定負荷状況データを補正する構成にあっては、推定負荷状況の推定精度が高まり、ひいては伸長率の推定精度、負荷分散の精度をより向上させることができる。

【００７９】

推定伸長率をベースとした各計算機の負荷指標として、推定伸長率そのものを負荷指標としたり、当該計算機へ新規にトランザクションを割当てる前あるいは割当て後における総推定伸長率（当該計算機における前記処理時間の推定伸長率に当該計算機の処理中トランザクション数を乗じた値）を用いたり、また、当該計算機へ新規にトランザクションを割り当てた後における前記総推定伸長率と、割当て前における前記総推定伸長率との差を用いる構成にあっては、個々の計算機のあるいはシステム全体としての負荷の程度を表現する負荷の指標に基づいた負荷配分が可能となり、個々のトランザクション自体の処理時間の最短化だけでなく、その割当てが他に及ぼす影響まで考慮した、システム全体としての最適化も可能となる。

【００８０】

本発明は、集中的に動的に個々の処理要求を配分する構成のシステムにも、静的／準静的に配分をされてしまった後で受けた計算機が配分の修正という位置づけで処理の転送を行うことになるという構成のシステムにも適用が可能である。シミュレーションを用いた性能評価結果が、前述のように図８、図９に示されて

いる。

【図面の簡単な説明】

【図１】

本発明の第１の実施の形態の構成を示す示すブロック図である。

【図２】

任意の計算機ｉから見たシステムのモデルである。

【図３】

負荷指標の値を計算するために用いるデータの一覧表である。

【図４】

（Ａ）第１の実施の形態における実行計算機選択手段の動作を示す流れ図である。

（Ｂ）第２あるいは第３の実施の形態における実行計算機選択手段の動作を示す流れ図である。

【図５】

本発明で用いる８種類の負荷の指標を示す一覧表である。

【図６】

本発明の第２の実施の形態の構成を示すブロック図である。

【図７】

本発明の第３の実施の形態の構成を示すブロック図である。

【図８】

平均応答時間（本方式のシミュレーション結果）のグラフである。

【図９】

応答時間の標準偏差（本方式のシミュレーション結果）のグラフである。

【符号の説明】

１　　計算機群

１０　　交換・蓄積機構

１１～１ｎ　　計算機１～ｎ

１ｘ１　　計算機ｘ上の負荷データ測定手段Ａ

１ｘ２　　計算機ｘ上のトランザクション処理手段

１ｘ３　　計算機ｘ上の通信手段

２　　中継配分装置

２１　　通信手段

２５　　中継仮配分装置

３　　高速チャネル

４　　ファイル装置群

５　　端末装置群

５１　　通信網

６　　負荷データ記憶手段

６ｘ　　計算機ｘ上の負荷データ記憶手段

７　　実行計算機選択手段

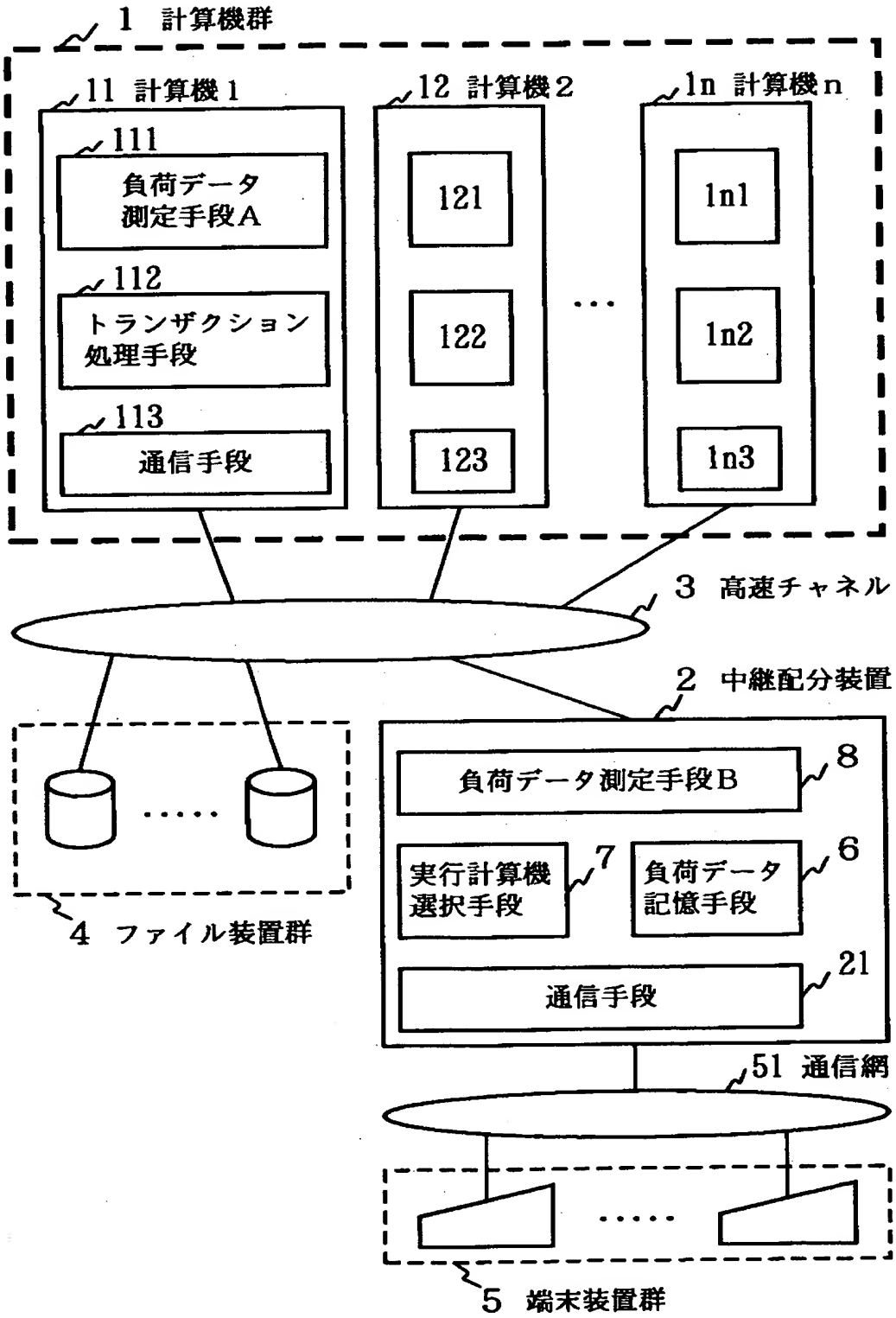
７ｘ　　計算機ｘ上の実行計算機選択手段

８　　負荷データ測定手段Ｂ

８ｘ　　計算機ｘ上の負荷データ測定手段Ｂ

【書類名】　　図面

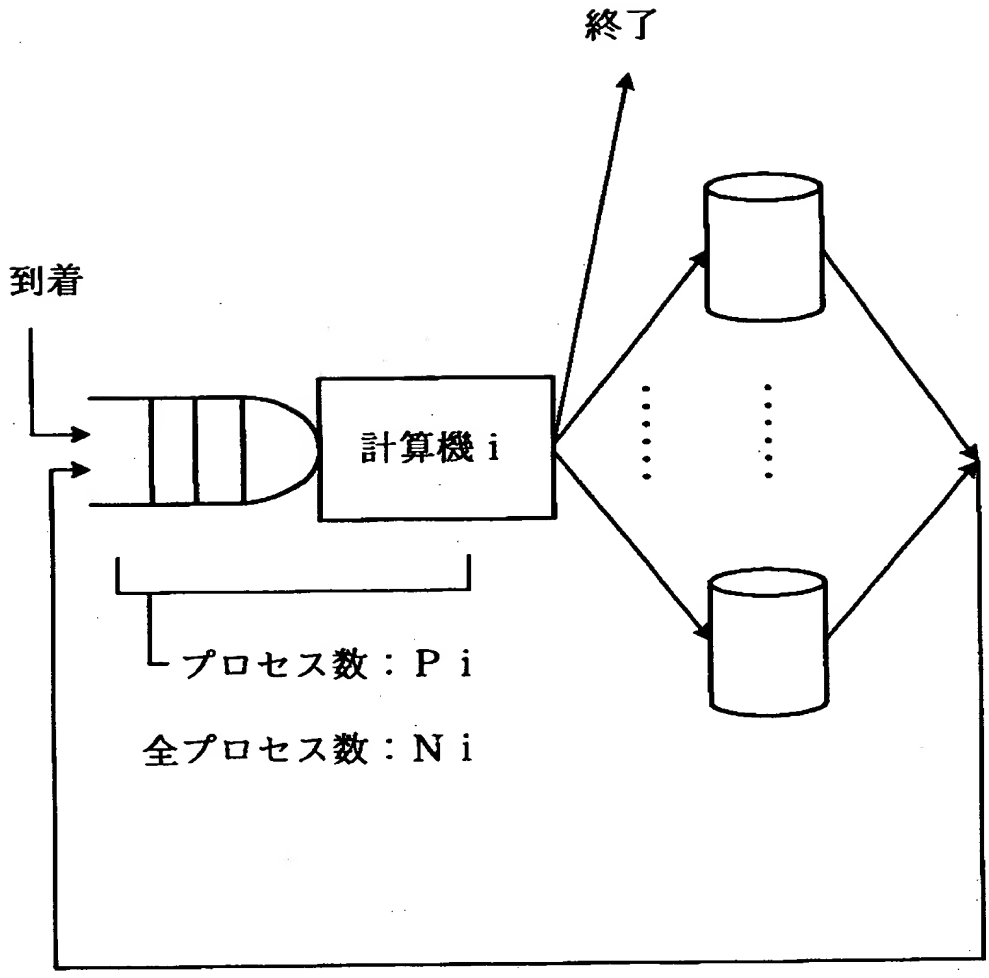
【図１】






出証特２０００－３０５５２１４

【図2】

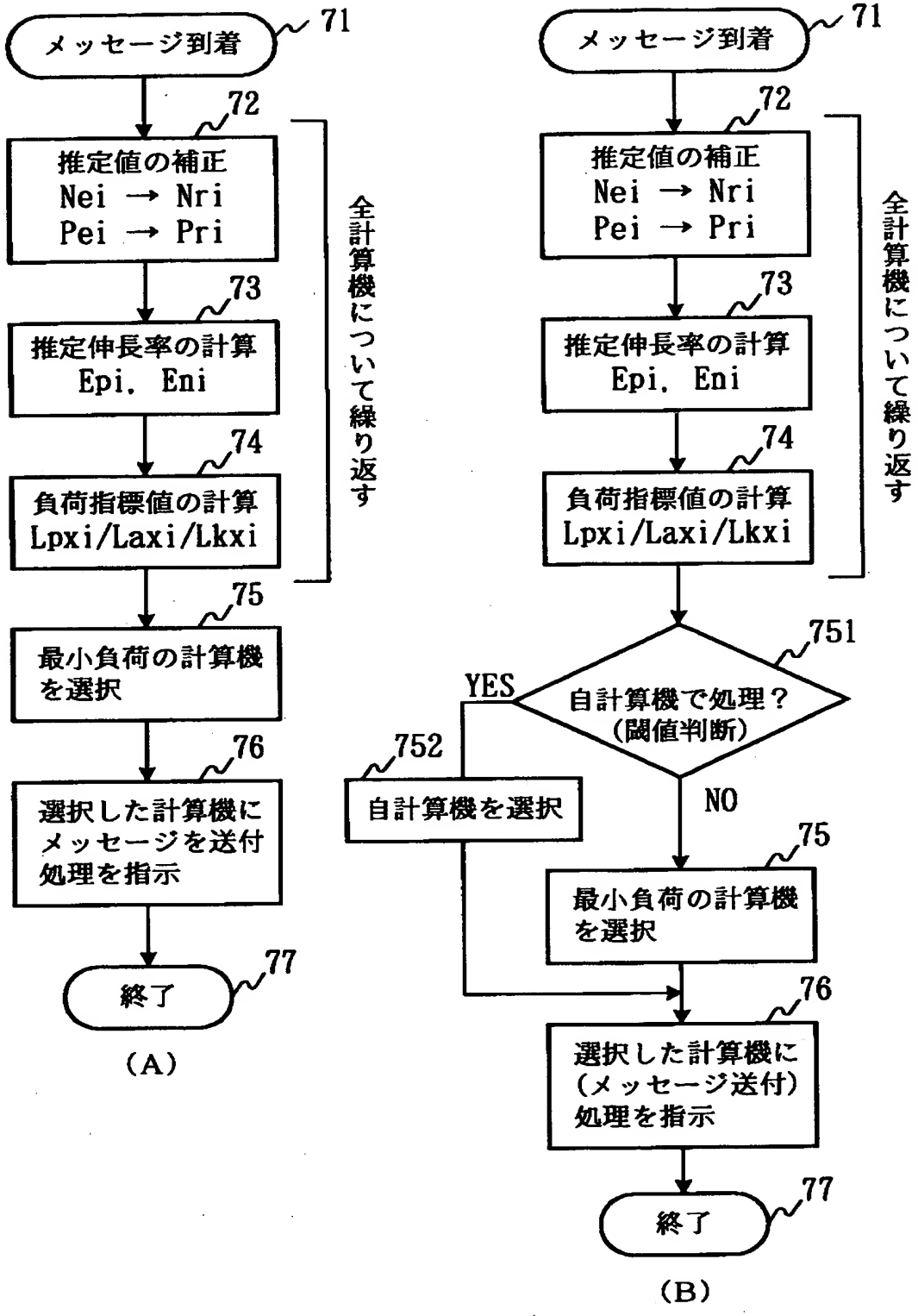


【図3】

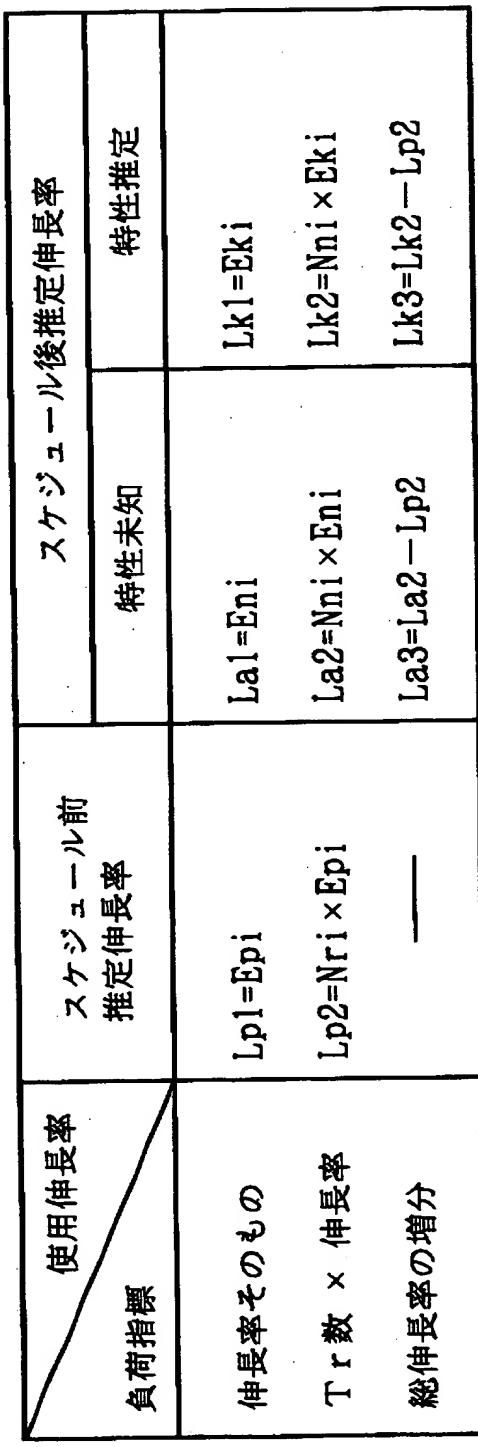
| T1 | T2 | T3 | T4 | T5 | T6 | T7 |
|---|---|---|---|---|---|---|
| 計算機# | 現ｔｒ数 | 推定ｔｒ数 | 推定ｔｒ数／CPU系 | 推定伸長率 | 配分後推定伸長率 | 推定負荷 |
| 1 | Np1 | Ne1 | Pe1 | Ep1 | En1 | Lxy1 |
| 2 | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ |
| i | Npi | Nei | Pei | Epi | Eni | Lxyi |
| ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ |
| n | Npn | Nen | Pen | Epn | Enn | Lxyn |

x＝p／a／k　　y＝1／2／3

【図4】

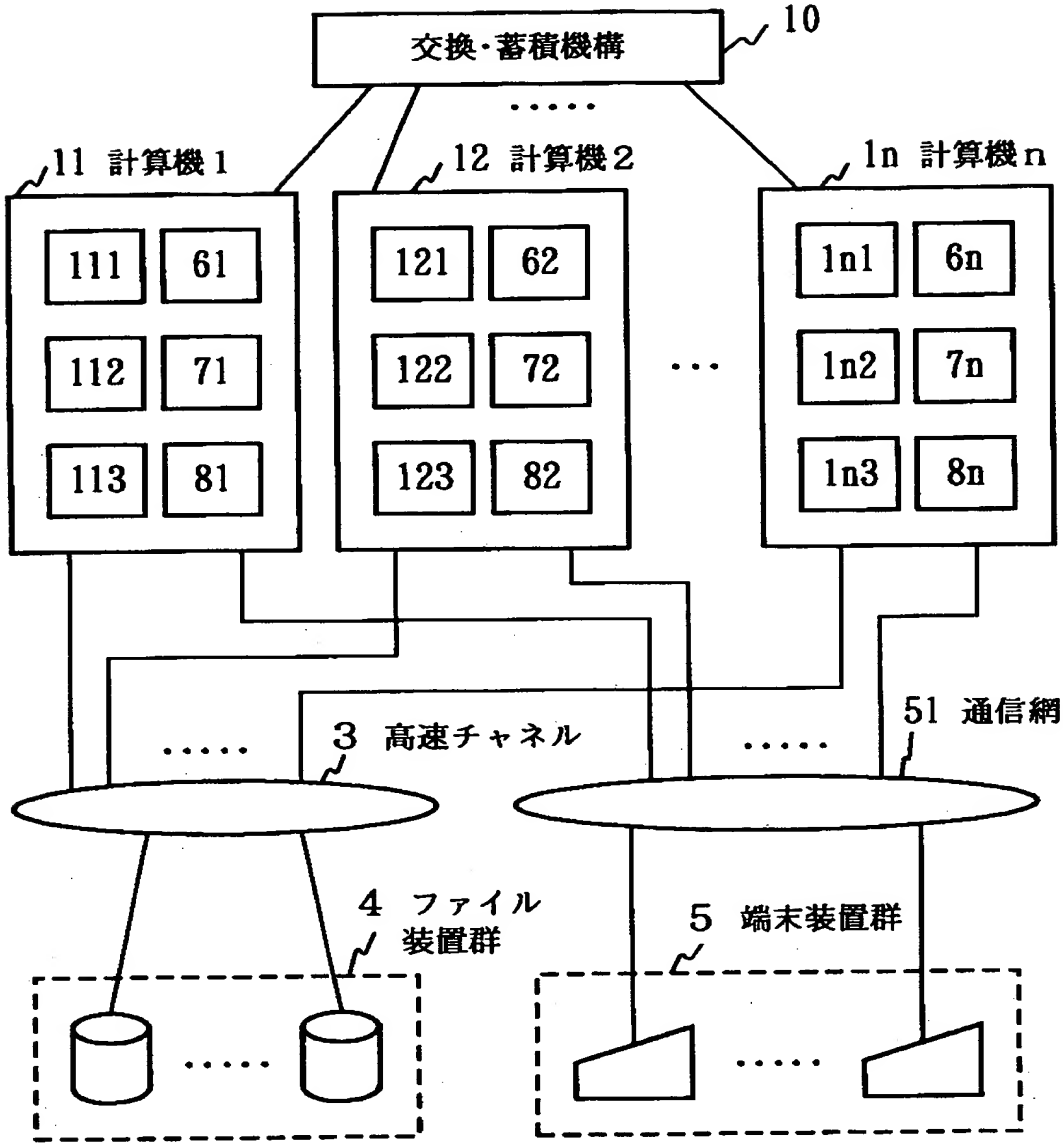

メッセージ到着
71
72
推定値の補正
Nei → Nri
Pei → Pri
73
推定伸長率の計算
Epi, Eni
74
負荷指標値の計算
Lpxi/Laxi/Lkxi
全計算機について繰り返す
75
最小負荷の計算機を選択
76
選択した計算機にメッセージを送付処理を指示
77
終了
(A)
メッセージ到着
71
72
推定値の補正
Nei → Nri
Pei → Pri
73
推定伸長率の計算
Epi, Eni
74
負荷指標値の計算
Lpxi/Laxi/Lkxi
全計算機について繰り返す
751
自計算機で処理？
(閾値判断)
YES
752
自計算機を選択
NO
75
最小負荷の計算機を選択
76
選択した計算機に(メッセージ送付)処理を指示
77
終了
(B)

【図5】



| 使用伸長率 / 負荷指標 | スケジュール前 推定伸長率 | スケジュール後推定伸長率 | |
|---|---|---|---|
| | | 特性未知 | 特性推定 |
| 伸長率そのもの | Lp1=Epi | La1=Eni | Lk1=Eki |
| Ｔｒ数 × 伸長率 | Lp2=Nri×Epi | La2=Nni×Eni | Lk2=Nni×Eki |
| 総伸長率の増分 | —— | La3=La2－Lp2 | Lk3=Lk2－Lp2 |

【図6】




61, 62, ･･･, 6n ； 負荷データ記憶手段

71, 72, ･･･, 7n ； 実行計算機選択手段

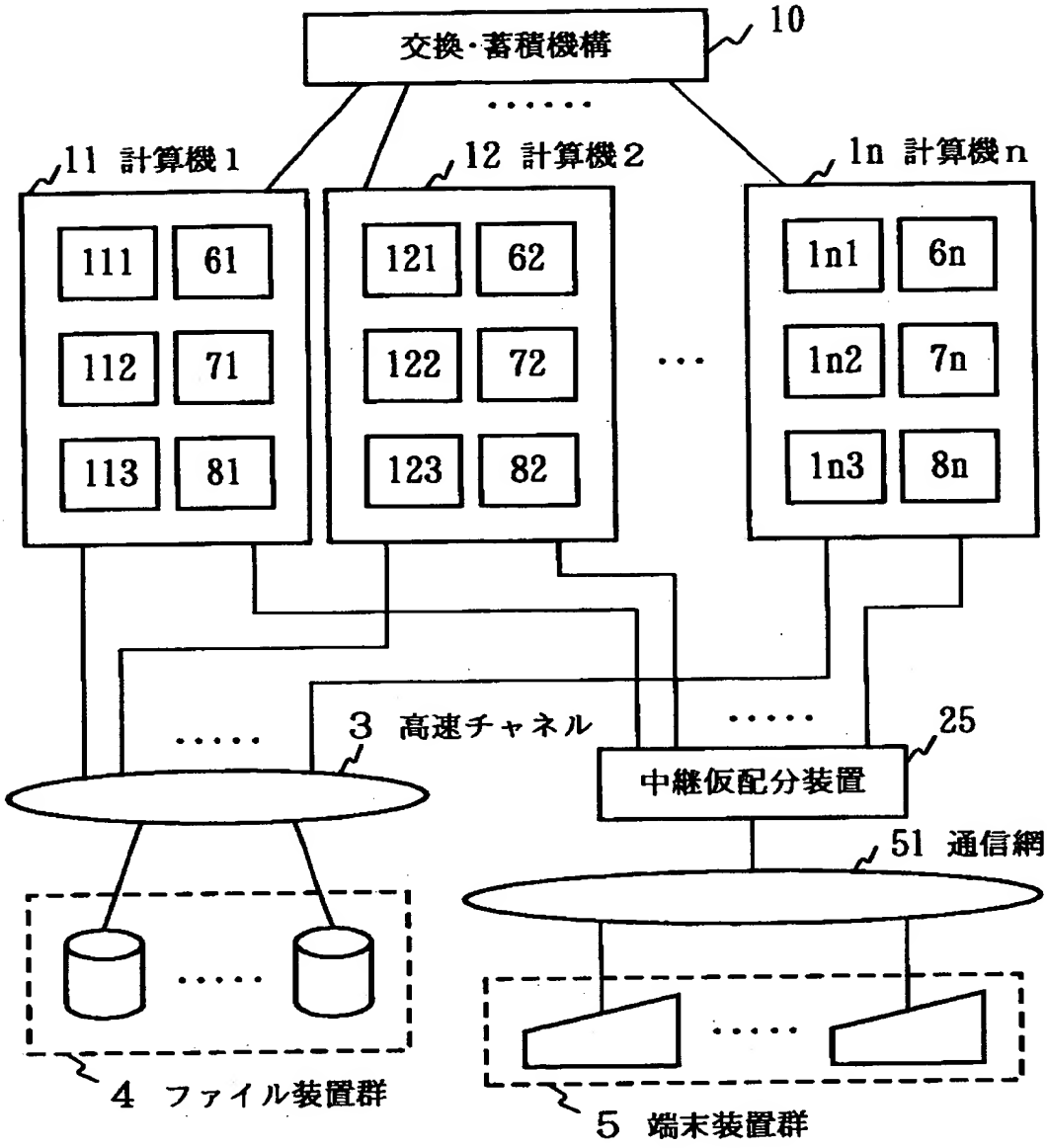
81, 82, ･･･, 8n ； 負荷データ測定手段Ｂ

111, 121, ･･･, 1n1 ； 負荷データ測定手段Ａ

112, 122, ･･･, 1n2 ： トランザクション処理手段

113, 123, ･･･, 1n3 ； 通信手段

【図7】




61, 62, ･･･, 6n ； 負荷データ記憶手段

71, 72, ･･･, 7n ； 実行計算機選択手段

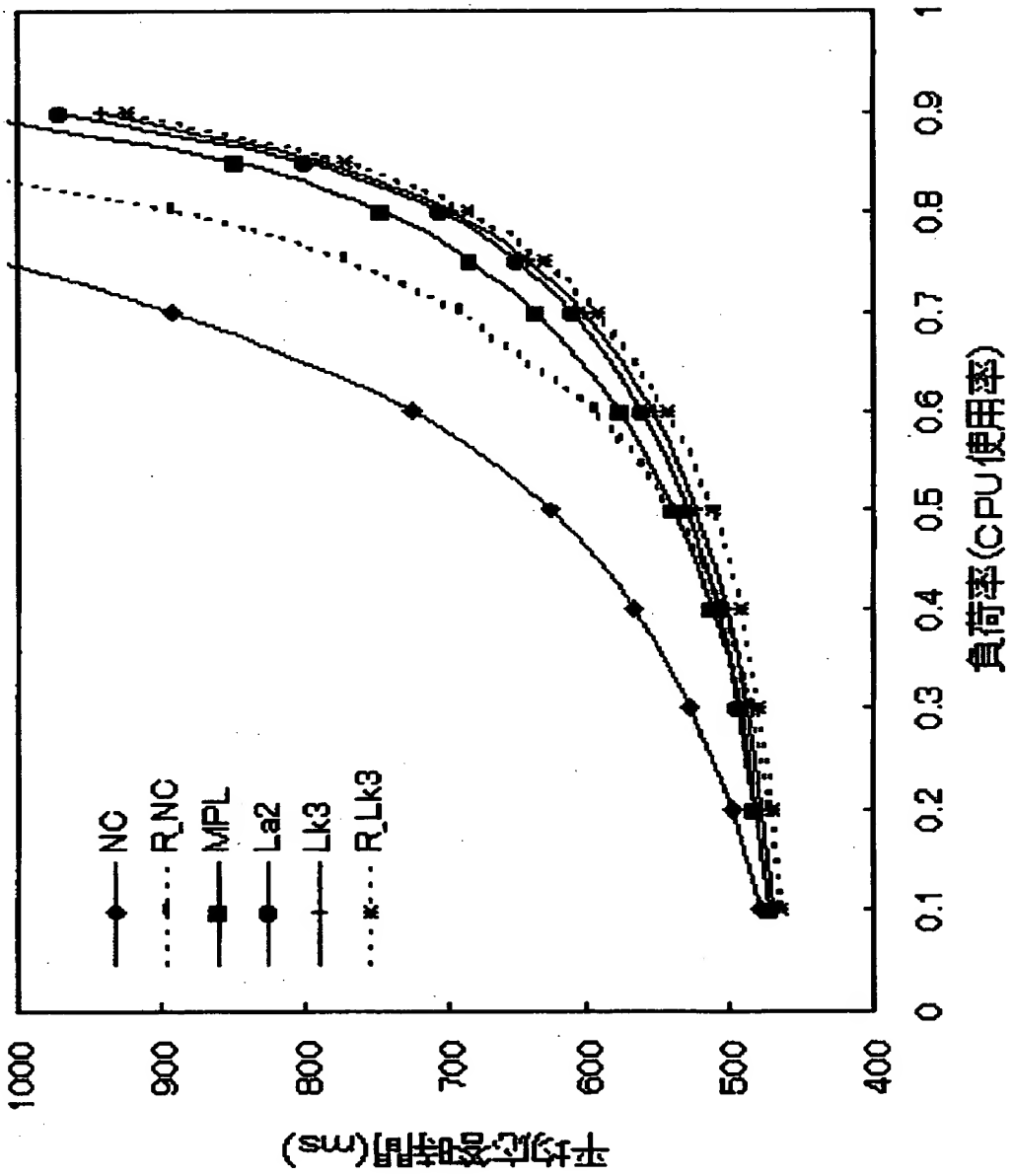
81, 82, ･･･, 8n ； 負荷データ測定手段Ｂ

111, 121, ･･･, 1n1 ； 負荷データ測定手段Ａ

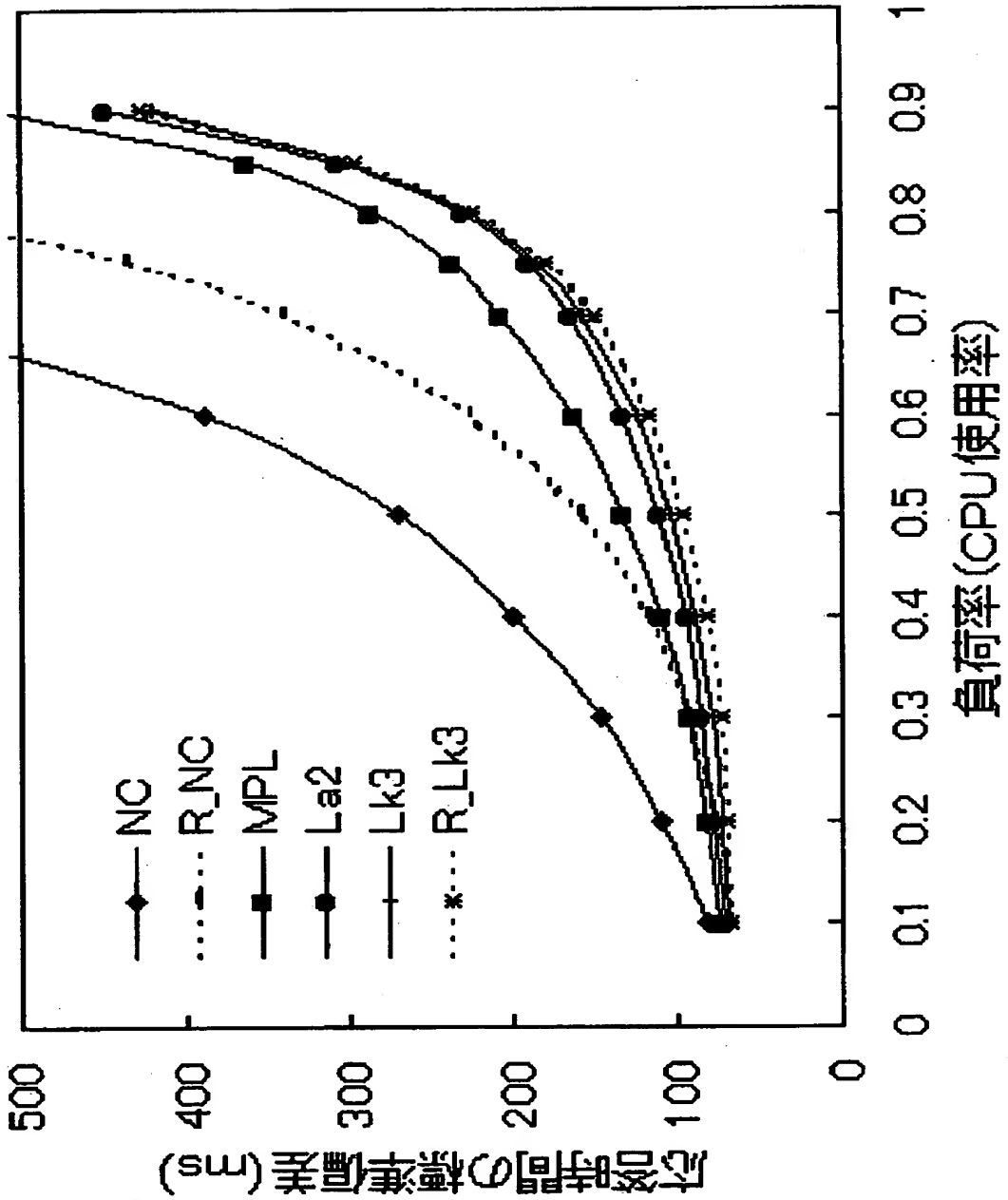
112, 122, ･･･, 1n2 ； トランザクション処理手段

113, 123, ･･･, 1n3 ； 通信手段

【図8】


NC
R_NC
MPL
La2
Lk3
R_Lk3
平均応答時間(ms)
1000
900
800
700
600
500
400
負荷率(CPU使用率)
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1

【図9】


応答時間の標準偏差(ms)
0
100
200
300
400
500
負荷率(CPU使用率)
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
NC
R_NC
MPL
La2
Lk3
R_Lk3

【書類名】　　要約書

【要約】

【課題】　トランザクション処理要求を発生する端末装置群と要求の処理を負荷分担型により実行する複数の計算機からなるシステムにおいて、各計算機の負荷を動的にバランスさせることにより、全体の応答性能を良くする。すなわち、応答時間の平均とばらつきを小さくする。

【解決手段】　一定時間ごとに各計算機の処理中トランザクション数、およびＣＰＵ系に滞在するプロセス数またはＣＰＵ使用率を測定し、また、正確な処理中トランザクション現在数を常に把握し、これらの測定値を元に各計算機における処理時間の伸長率を推定し（７３）、この推定伸長率に基づいて負荷指標の値を求め（７４）、負荷の低い計算機へトランザクションの配分を行う（７５、７６）。各計算機上で実行中のジョブミックスの特性まで考慮した動的配分が実現でき、応答性能を上げられる。

【選択図】　図４

# 出　願　人　履　歴　情　報

識別番号　　〔０００００４２３７〕

１．変更年月日　　１９９０年　８月２９日
［変更理由］　新規登録
住　所　　東京都港区芝五丁目７番１号
氏　名　　日本電気株式会社